

NAVAL POSTGRADUATE SCHOOL
Monterey, California

AD-A267 410



DTIC
ELECTE
AUG 4 1993
S C D

THESIS

**A FORMAL PROTOCOL TEST PROCEDURE
FOR THE SURVIVABLE ADAPTABLE FIBER
OPTIC EMBEDDED NETWORK
(SAFENET)**

by

Wayne High

March 1993

Thesis Advisor:

G. M. Lundy

Approved for public release; distribution is unlimited.

93-17362



11/6/93

93 8 3 0 8 3

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION AVAILABILITY OF REPORT Approved for public release; distribution is unlimited	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S)	
6a. NAME OF PERFORMING ORGANIZATION Computer Science Dept. Naval Postgraduate School	6b. OFFICE SYMBOL (if applicable) CS	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School	
6c. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000		7b. ADDRESS (City, State, and ZIP Code) Monterey, CA 93943-5000	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (if applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER	
8c. ADDRESS (City, State, and ZIP Code)		10. SOURCE OF FUNDING NUMBERS	
		PROGRAM ELEMENT NO	PROJECT NO
		TASK NO	WORK UNIT ACCESSION NO.
11. TITLE (Include Security Classification) A FORMAL PROTOCOL TEST PROCEDURE FOR THE SURVIVABLE ADAPTABLE FIBER OPTIC EMBEDDED NETWORK (SAFENET) (U)			
12. PERSONAL AUTHOR(S) Wayne High			
13a. TYPE OF REPORT Master's Thesis	13b. TIME COVERED FROM 03/91 TO 03/93	14. DATE OF REPORT (Year, Month, Day) March 1993	15. PAGE COUNT 120
16. SUPPLEMENTARY NOTATION The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the United States Government.			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB-GROUP	
		SAFENET profile specification, conformance testing	
19. ABSTRACT (Continue on reverse if necessary and identify by block number) This thesis focuses upon a new method for verifying the correct operation of a complex, high speed fiber optic communication network. These networks are of growing importance to the military because of their increased connectivity, survivability, and reconfigurability. With the introduction and increased dependence on sophisticated software and protocols, it is essential that their operation be correct. Because of the speed and complexity of fiber optic networks being designed today, they are becoming increasingly difficult to test. Previously, testing was accomplished by application of conformance test methods which had little connection with a implementation's specification. The major goal of conformance testing is to ensure that the implementation of a profile is consistent with its specification. Formal specification is needed to ensure that the implementation performs its intended operations while exhibiting desirable behaviors. The new conformance test method presented is based upon the System of Communicating Machine model which uses a formal protocol specification to generate a test sequence. The major contribution of this thesis is the application of the System of Communicating Machine model to formal profile specifications of the Survivable Adaptable Fiber Optic Embedded Network (SAFENET) standard which results in the derivation of test sequences for a SAFENET profile. The results of applying this new method to SAFENET's OSI and Lightweight profiles are presented.			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED/UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL G. M. Lundy		22b. TELEPHONE (Include Area Code) (408) 656-2094	22c. OFFICE SYMBOL CS/37

Approved for public release; distribution is unlimited

**A FORMAL PROTOCOL TEST PROCEDURE
FOR THE SURVIVABLE ADAPTABLE FIBER
OPTIC EMBEDDED NETWORK
(SAFENET)**

by
Wayne High
Lieutenant, United States Navy
B.S. Computer Science, University of South Carolina, 1986

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF COMPUTER SCIENCE


from the

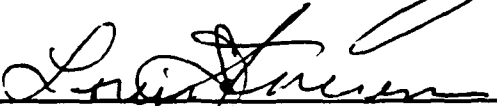
NAVAL POSTGRADUATE SCHOOL
March 1993

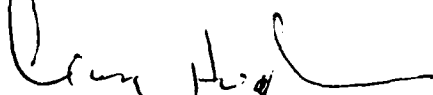
Author:


Wayne High

Approved By:

 9 Mar 1993
G. M. Lundy, Thesis Advisor

 9/1 Mar 1993
Louis Stevens, Second Reader


CDR. Gary J. Hughes, Chairman,
Department of Computer Science

ABSTRACT

This thesis focuses upon a new method for verifying the correct operation of a complex, high speed fiber optic communication network. These networks are of growing importance to the military because of their increased connectivity, survivability, and reconfigurability. With the introduction and increased dependence on sophisticated software and protocols, it is essential that their operation be correct. Because of the speed and complexity of fiber optic communication networks being designed today, they are becoming increasingly difficult to test. Previously, testing was accomplished by application of conformance test methods which had little connection with an implementation's specification. The major goal of conformance testing is to ensure that the implementation of a profile is consistent with its specification. Formal specification is needed to ensure that the implementation performs its intended operations while exhibiting desirable behaviors. The new conformance test method presented is based upon the System of Communicating Machine model which uses a formal protocol specification of the implementation to generate a test sequence.

The major contribution of this thesis is the application of the System of Communicating Machine model to formal profile specifications of the Survivable Adaptable Fiber Optic Embedded Network (SAFENET) standard which results in the derivation of test sequences for a SAFENET profile. The results of applying this new method to SAFENET's OSI and Lightweight profiles are presented.

DTIC QUALITY INSPECTED 3

Accession For	
NTIS CRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Doc	Avail and/or Special
A-1	

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	PURPOSE.....	1
B.	BACKGROUND.....	2
C.	CONTENTS.....	3
II.	FDDI BACKGROUND.....	4
III.	SAFENET BACKGROUND.....	9
A.	SERVICE PARTITIONS.....	9
B.	PROTOCOL SUITES.....	10
C.	SAFENET TOPOLOGY.....	16
D.	SAFENET FIBER OPTIC DEVELOPMENT.....	16
IV.	THE SAFENET STANDARD.....	19
A.	THE SAFENET STANDARD AND ITS TESTABILITY.....	19
B.	PROBLEMS WITH THE CURRENT SAFENET STANDARD.....	20
V.	TESTING FIBER OPTIC LINKS.....	25
A.	LOSS BUDGET.....	26
B.	SYSTEM THROUGHPUT.....	27
1.	Clock Accuracy Verification.....	28
2.	Timing Test Procedure.....	28
VI.	TESTING SAFENET'S IMPLEMENTATION PROFILES.....	30
A.	TEST METHOD.....	30
1.	Preliminary Steps.....	31
2.	Test Sequence Generating Procedure.....	32
3.	Refining Steps.....	34
B.	APPLICATION OF TEST METHOD.....	35
1.	OSI Profile Specification.....	35
2.	OSI Test Sequence Generation.....	43
a.	Preliminary Steps.....	43
b.	Test Sequence Generation.....	44
3.	Lightweight Profile Specification.....	50
4.	Lightweight Test Sequence Generation.....	56
a.	Preliminary Steps.....	56
b.	Test Sequence Generation.....	57
VII.	CONCLUSIONS AND RECOMMENDATIONS.....	61
A.	CONTRIBUTIONS OF THIS RESEARCH.....	61
B.	AREAS FOR FURTHER RESEARCH.....	62
	APPENDIX.....	63
	LIST OF REFERENCES.....	111
	INITIAL DISTRIBUTION LIST.....	113

LIST OF TABLES

Table 1, OSI PREDICATE-ACTIONS	42
Table 2, TEST SEQUENCE FOR THE OSI PROFILE	46
Table 3, LIGHTWEIGHT PREDICATE-ACTIONS.....	52
Table 4, TEST SEQUENCE FOR THE LIGHTWEIGHT PROFILE	58

LIST OF FIGURES

Figure 1, Typical FDDI configuration.....	5
Figure 2, FDDI Frame Format.....	7
Figure 3, SAFENET Protocol Profile.....	8
Figure 4, Overview of the OSI Profile.....	12
Figure 5, Overview of the Lightweight Profile	14
Figure 6, Overview of the Combined Profile	15
Figure 7, An overview of component systems comprising SAFENET.....	17
Figure 8, OSI Profile Specification	38
Figure 9, OSI Network Node Data Structure.....	41
Figure 10, Lightweight Profile Specification	53
Figure 11, Lightweight Network Node Data Structure	55

I. INTRODUCTION

A. PURPOSE

The Survivable Adaptable Fiber Optic Embedded Network (SAFENET) is part of the United States Navy's Next Generation Computer Resources (NGCR) program and represents an effort to meet the data transfer demands of current and future naval shipboard mission critical computer systems through development of standard computer network profiles. SAFENET represents a Local Area Network (LAN) grouping of standards, encompassing the seven ISO/OSI model layers. SAFENET is a program being researched and developed by a joint Navy-Industry working group to formulate standard commercial network methods to fit the requirements of the Navy's various combat platforms. Whenever possible, the joint working group intends to select well developed industry standards. The Navy's NGCR team is chartered to investigate methods for future shipboard hardware and software development to meet the Navy's requirements of survivability, increased connectivity, performance and future system growth capabilities [GREE89] [HDBK92]. SAFENET is an effort to meet the needs of current and future systems used aboard the Navy's combat ships, submarines, and aircraft.

Over the course of time, network designers learned that ambiguous rules can trigger undesirable sequences of events which can have adverse effects on the best design; therefore, it is essential that communication networks and their protocols be adequately tested. Entire networks, with possibly hundreds or even thousands of attached computers can be rendered essentially useless by a faulty protocol or profile. With every passing day our society is becoming more dependent on communication networks and their protocols. Consequently, with so many lives and costly equipment at stake it is imperative to test these profiles and protocols to ensure that they perform as intended. This thesis presents a conformance test method that is based upon earlier work [MILL90]. In this thesis, the conformance test method utilized is based upon a formal specification. In addition,

applications of the test method using real world SAFENET profiles as examples will be demonstrated.

B. BACKGROUND

Currently, in the Navy, computers are usually found configured in point-to-point interfaces. However, the growing trend of distributed architectures in modern naval combat systems requires a greater degree of system integration than present point-to-point interfaces can support. As a result, the SAFENET standards are being developed to solve communications connectivity and system integration problems by providing the shipboard computers with the capability to communicate with multiple devices and application programs over a single Input/Output port through the use of a computer network and to provide future system growth capacity.

To ensure that the elements of a complex communications network such as SAFENET communicate reliably once the system has been implemented, the protocols of the system should be verified against all system design requirements. In this manner, instances of incompleteness in a protocol specification can be located using protocol verification techniques. Provided formal specifications have been done, conformance testing is an essential step to ensuring accomplishment of intended functions without error. Effectively testing protocols with other software and hardware systems presents a difficult problem. Conformance testing's major goal is to ensure that the implementation of the protocol is consistent with its specification. Therefore, it is highly advantageous that the specification be expressed in a formal model that has been formally verified. A recent paper by Miller [MILL90], pointed out the developing rift between specification and verification, and between these two and conformance testing. Protocol models, designed for specification purposes, tend to have powerful program language constructs, which simplify specification but leads to a higher degree of verification and analysis difficulty. The Communicating Finite State Machine (CFSM) model is too simple for the specification of modern, complex

protocols because this protocol model is designed primarily with analysis in mind [LUND91].

C. CONTENTS

This thesis attempts to bridge the gap between SAFENET specification and testing. Assuming that all SAFENET protocols are not available in a form convenient for testing, simplifying the difficulty associated with verification, analysis and testing, we start from a protocol model called Systems of Communicating Machines (SCM). A procedure is presented for the generation of a test sequence for a protocol specified in terms of the SAFENET model. Then, the SAFENET procedure is used to generate a test sequence for a SAFENET protocol implementation.

The testing of any complex software is known to be a difficult problem, and this certainly applies to the testing of SAFENET protocols. Because SAFENET protocols are critical to so many systems, it is a problem which cannot be avoided or ignored. The procedure presented in this thesis does not detect all errors or combinations of errors. Only exhaustive testing can accomplish this, but substantial resources are required. The approach does, however represent an attempt to exercise a portion of a SAFENET protocol machine, thereby providing some assurance that the implementation fulfills its purpose.

Presently, there are two standards under development SAFENET I and SAFENET II. This thesis focuses on SAFENET II which uses the American National Standards Institute (ANSI) Fiber Distributed Data Interface (FDDI) LAN that operates at 100 Mbps. In the following chapter the FDDI standard is discussed. In Chapter III, the SAFENET Draft Standard is discussed followed by Chapter IV, which discusses problems found in the SAFENET Draft Standard. In Chapter V, the testing of fiber optic links is discussed. In Chapter VI, the test procedure is applied to the SAFENET protocol. The final chapter summarizes the thesis.

II. FDDI BACKGROUND

The FDDI standard is focused on the comprehensive implementation of communications through fiber optics; with fiber optics, information is passed through modulated beams of light on glass fiber rather than the more antiquated electronic pulses on copper wire. In large backbone Local Area Networks (LANs), FDDI has several advantages over copper wire. Fiber is oblivious to lightening strikes, to the Electro-Magnetic Pulse (EMP) phenomenon associated with nuclear detonations and to their resultant current surges that can damage connected equipment and cause safety hazards. Furthermore, fiber is not subject to radio or Electro-Magnetic Interference (EMI) and is generally less restrictive in its environment, requiring far less space in existing cable runs.

FDDI utilizes two counter-rotating token rings. The typical FDDI configuration has the primary ring carrying data and the secondary ring being used for automatic bypass and recovery (Figure 1). Current networks in service are generally comprised of CSMA/CD protocols, which limit transmission rates to 10 Mbps, shorter cable runs, and rapidly decreasing efficiency as the network load increases because of data collision resolution. The use of a ring topology offers the advantage of data collision elimination, which provides for very high effective data rates. Unlike CSMA/CD topologies (such as Ethernet), FDDI's performance does not degrade significantly with increased levels of traffic, up to 95% of its rated capacity. FDDI networks can bypass hardware failures. When a node or link fails, the two counter rotating paths wrap together around the fault thus allowing continued communications. Any fault on FDDI dual rings can be isolated, keeping the remainder of the rings completely active. When the fault is corrected the fiber optic ring reconfigures automatically. This ability to adapt to breaks or node failures helps ensure reliability of the system and data availability in transfer.

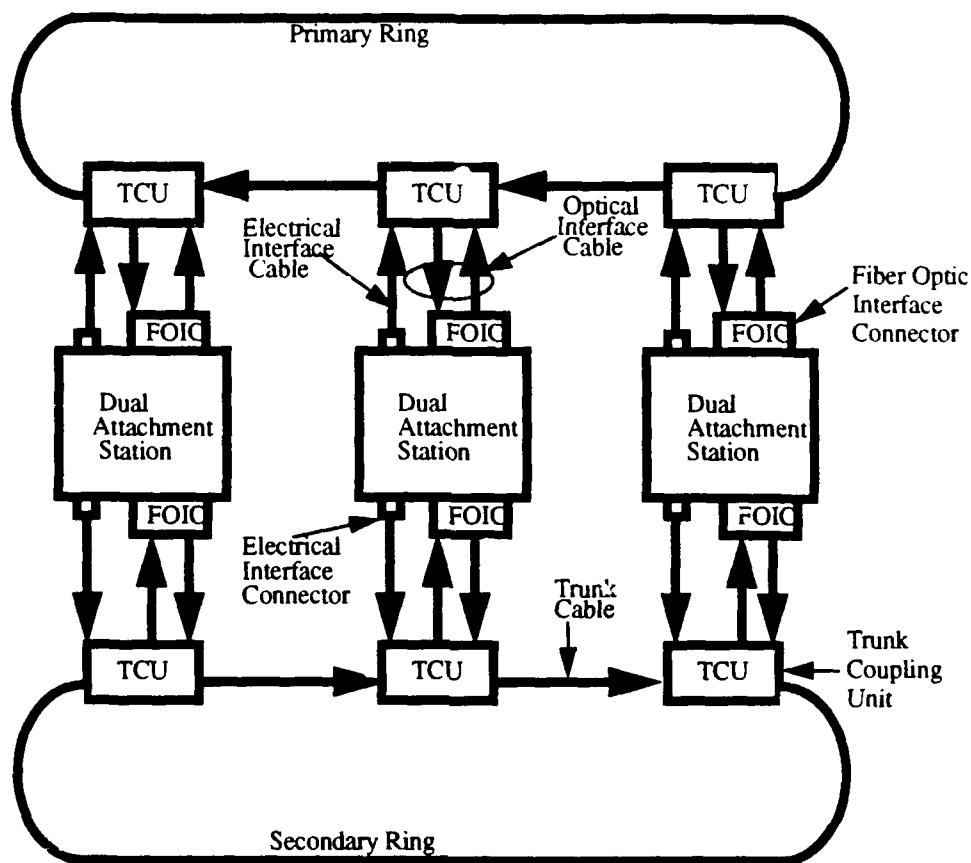


Figure 1 Typical FDDI configuration

A special bit pattern, called a token, is continuously circulated by the FDDI ring. Stations transmit data by capturing the token, transmitting their traffic, and sending the token to the next station on the network until a complete circuit is made. FDDI supports two types of traffic transmissions: synchronous and asynchronous. Synchronous service is designed for applications with predictable bandwidths and critical response times. Asynchronous services are designed for applications with bursty, widely varying bandwidth requirements. To accommodate asynchronous, "non-deterministic" traffic a Target Token Rotation Time (TTRT) is defined and negotiated during ring initialization. Each station maintains the same value for TTRT. Each station's Token Rotation Timer (TRT) is initialized to TTRT when enabled. The TRT counts down until $TRT = 0$ and is then reset to TTRT. The variable Late Count (LC) is initialized to zero ($LC = 0$) and is

incremented each time TRT reaches zero. In this manner, the Late Count counter records the number of times TRT has expired since the token was last received by a designated station. If TRT does not reach zero and LC is zero, the token is considered to have arrived early. When a station receives the token early, after transmitting any synchronous frames, it may transmit asynchronous frames for a period not to exceed the remaining TRT. Once the allotment of asynchronous frames are transmitted, the token is passed to the next station and both TRT and LC are reinitialized [STAL88]. Synchronous traffic is "deterministic" because each station is guaranteed token service within a specified time limit and for a specified allocation. In the event that a station has asynchronous, "non-deterministic" traffic to transmit, upon receipt of the token the station, first, transmits any synchronous traffic up to its allocation; then, if there is time remaining as the result of an early token arrival due to decreased synchronous allocation usage or if excess bandwidth is present, the asynchronous traffic will be transmitted.

Utilizing FDDI as the backbone LAN offers other advantages. In addition to satisfying the need of connecting LANS together without compromising inter-LAN speed, FDDI offers capability that will enable future technologies, including circuit switching. Like most networks, FDDI is a packet switched network, utilizing FDDI packets to facilitate the efficient transmission of data in various sizes. FDDI frames vary in length, have their own delimiting start and end markers, and contain their own destination addresses (Figure 2). By using an upward compatible extension of FDDI known as FDDI-II, FDDI gains the capability to perform circuit-switched service in addition to normal packet switching [ROSS89]. This permits future special applications which require real-time response from the network, including digital voice; rapid updating of tactical displays in battle may be another application. Ross describes the circuit switched connection as a "data stream," which provides for the transmission of continuous (analog) data.

Preamble	Starting Delimiter	Frame Control	Desti- nation Address	Source Address	Routing Infor- mation	Information Field	Frame Check Sequence	Ending Delimiter	Frame Status
64bits	8bits	8bits	16 or 48 bits	16 or 48 bits		< 4500octets	32bits	4bits	4bits

Figure 2 FDDI Frame Format

FDDI rings support two types of stations: dual attach stations (DAS), which attach directly to the ring, and single attach stations (SAS), such as PC's and work stations. Each DAS has four fiber connections, two to receive and transmit on the primary ring, and two to receive and transmit on the secondary ring. A typical DAS can be a concentrator, bridge, router, server, minicomputer or mainframe. Multiple DASs are linked together to form the network backbone. A SAS can be immediately isolated in case of fault detection without disrupting traffic on the ring.

In a distributed network environment, all the DASs in a FDDI ring participate in network capability, fault recovery, management, and network initialization. Internal DAS timers and logic control resolution of all ring failures provide bypass handling. Therefore, the counter-rotating ring topology allows the network to continue functioning in the event that either a node or link is lost. It is this survivability, in addition to its very high bandwidth, that makes FDDI most suitable to a battle environment. Optical fiber minimizes interference and signal degradation, and minimizes signal loss over long cable runs. Due to the extremely large bandwidth of fiber, bit-serial transmission may be used, offering the advantage of hardware simplicity, decreased complexity, and increased reliability [ROSS89]. Reliability is a key factor for the Navy in both normal peacetime operations and while at battle. This interest in reliability and communications connectivity led to the development effort of Survivable Adaptable Fiber Optic Embedded Network (SAFENET).

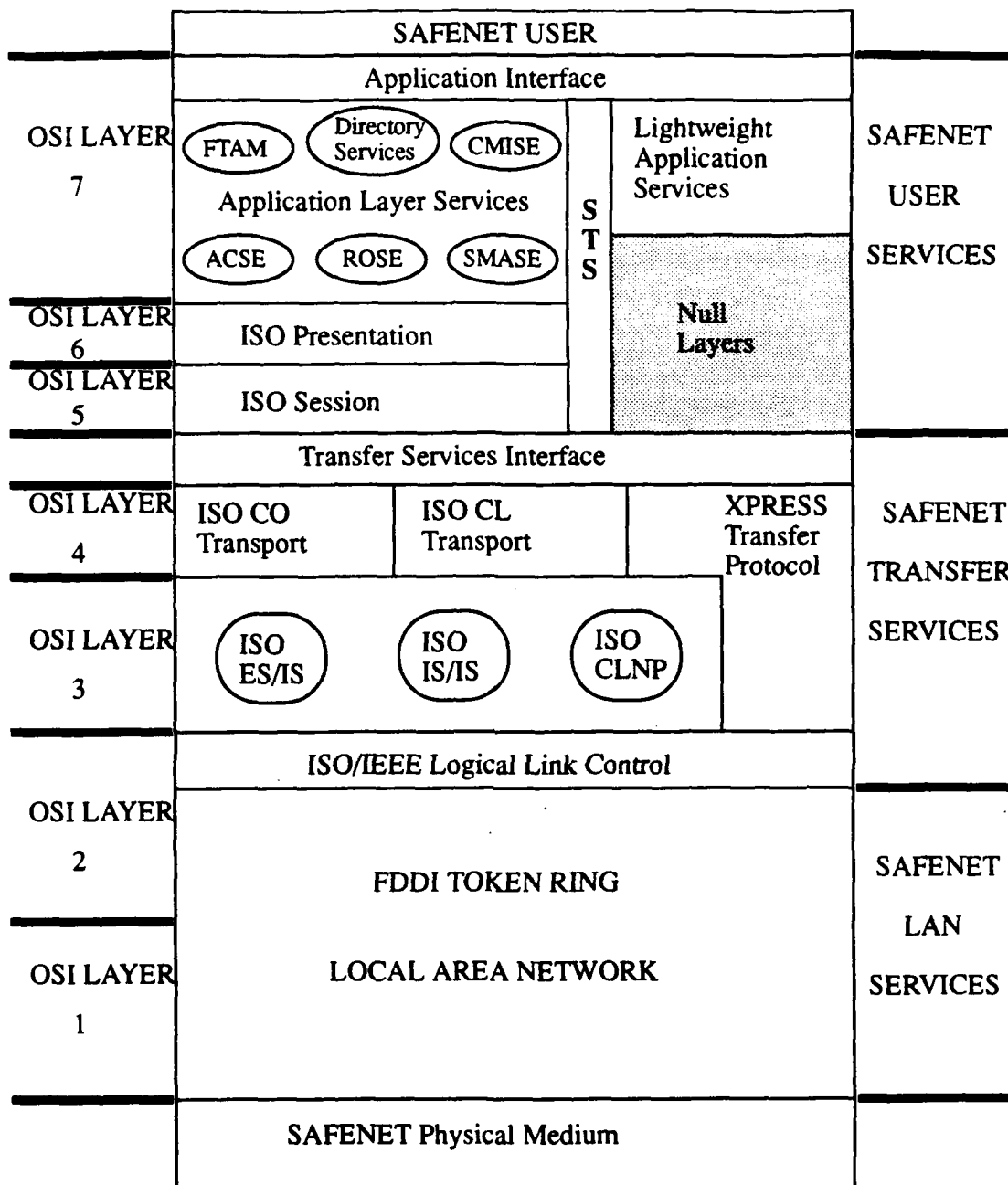


Figure 3 SAFENET Protocol Profile

III. SAFENET BACKGROUND

SAFENET is based on the FDDI token-ring standard and incorporates profiles for both ISO compatibility and real time performance. By employing the seven layer ISO reference model for Open Systems Interconnection (OSI), SAFENET specifies protocols at each layer of this model, defining the complete SAFENET profile. In SAFENET, this protocol profile is organized in two ways: by service partitions and by defined profiles.

A. SERVICE PARTITIONS

The first method of SAFENET's communicating architecture divides the protocol profile into three service partitions: user services, transfer services, and LAN services. Each of these partitions encompasses a portion of the seven layers of the ISO reference model. Figure 3 delineates the seven layers of the ISO reference model on the left, the major elements of the SAFENET profile in the center, and the service partitions on the right. The user services partition corresponds to the session, presentation, and application layers of the ISO reference model (layers 5-7) and is that portion of the SAFENET profile in which users interact with the network. The user services partition afford SAFENET users with the capability to interact with, manage, and respond to the underlying transfer services. In the center of the SAFENET profile lies the transfer services partition. It corresponds to the network, transport and Logical Link Control (LLC) sublayer of the Data Link Layer, (layers 2-4) of the ISO reference model. Through these services reliable communication mechanisms are provided to SAFENET users. The LAN services partition is that part of the SAFENET profile which performs the actual data transfer and corresponds to the physical layer of the ISO reference model as well as the media access control sublayer of the data link layer (layers 1-2). The LAN services consist of the upload and download ability to get data on and off the physical medium in a controlled manner.

B. PROTOCOL SUITES

The second descriptive method of SAFENET communications architecture separates the protocols into defined profiles: the OSI protocol profile, lightweight protocol profile, and the combined protocol profile. As depicted in figures 4, 5, and 6, these profiles define the three distinct implementation classes permitted in SAFENET. It is not required that all stations on a given SAFENET network implement the same profiles. Each respective profile describes a specific combination of network protocols defined within SAFENET. When designing a SAFENET station, at least one of the profiles (OSI, Lightweight, and Combined profiles) must be implemented. However, network designers must ensure the presence of sufficient profiles at each station to ensure that the system meets its designed communications connectivity. Some of the services and protocols are common to all implementation classes and others are used only in the OSI or lightweight profiles. The SAFENET Time Service (STS) is required for all protocol suite implementations.

The OSI protocol suite, possessing protocols and services based upon ISO standards, provides complete OSI compliant networking services to systems which require it. The OSI protocol suite consists of Manufacturing Automation Protocol (MAP) private communications, ISO File Transfer, Access and Management (FTAM), Directory Services, Association Control Service Element (ACSE), Remote Operations Service Element (ROSE), System Management Application Service Element (SMASE), Common Management Information Service Element (CMISE), presentation and session layers, Connection-Oriented Transport Protocol (COTP), ISO Connectionless Transport Protocol (CLTP) which allows the user to transmit a single unit of data, datagram, without establishing a connection, ISO Connectionless Network Protocol (CLNP) which provides services for network routing and for the segmentation and reassembly of transport layer messages that are too large for the underlying communications service, ES/IS routing exchange protocol which provides stations with the ability to associate a data link layer address with a given network layer address, IS/IS intra-domain routing protocol which dynamically determines routes to pass data between intermediate systems, LLC type 1

(connectionless) protocol and the FDDI protocols. The ISO connection oriented Transport protocol class 4 (TP4) is required within the transfer services partition [ISO870]. This is done to ensure interoperability in an open systems environment. The OSI protocol suite is basically required when either the interoperability of independently developed nodes is a driving consideration, or the file handling capabilities of FTAM are required, or increased complexity requires network management; however, it adds this capability at the expense of delayed data flow and inability to supply multiple users simultaneously [KOCH91] [PAIG90]. Figure 4 shows an overview of the OSI protocol suite.

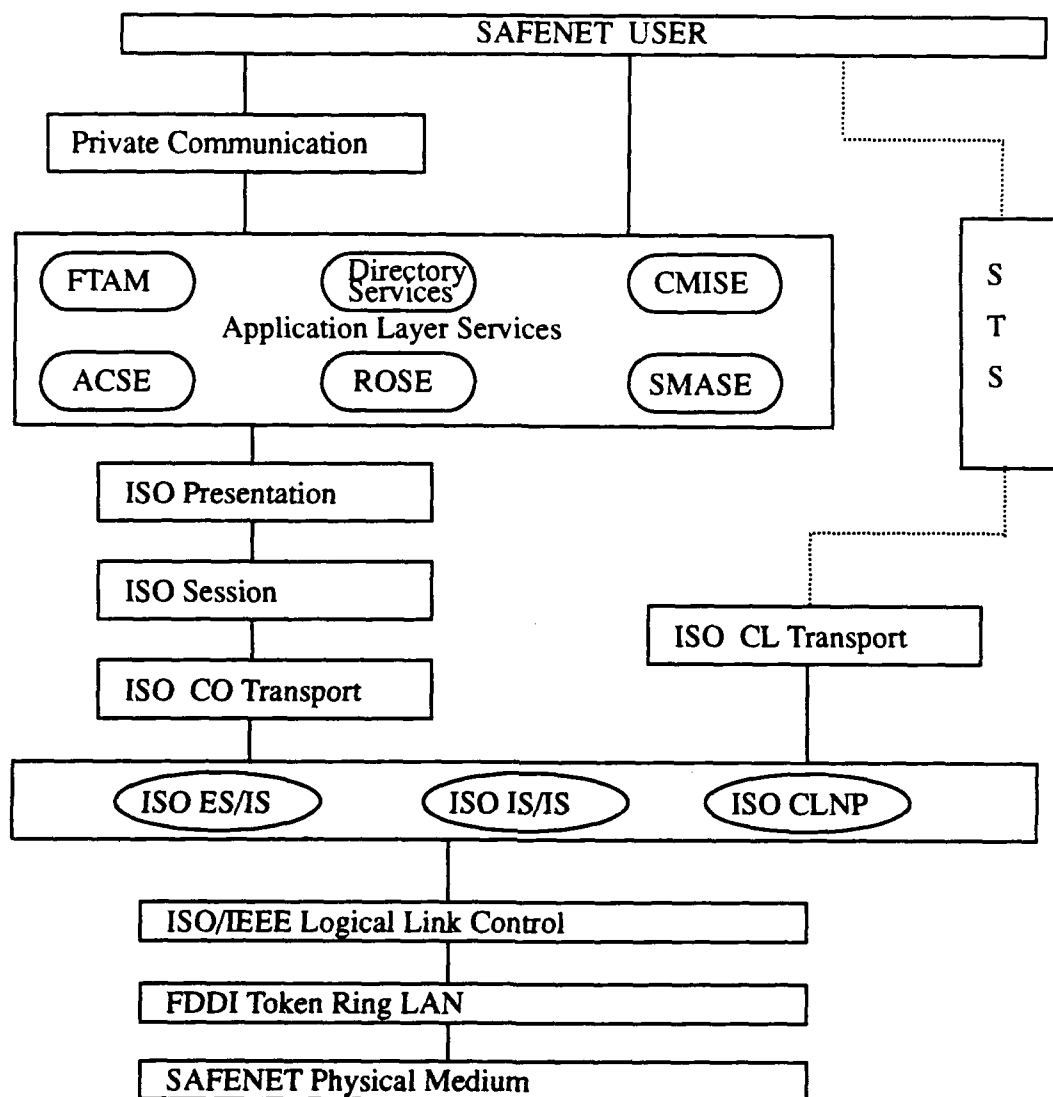


Figure 4 Overview of the OSI Profile

In circumstances where control of timing is critical from a resource point of view, the lightweight protocol suite provides process to process message passing services. The message passing services support point to point, multicast, and remote procedure call (transaction) styles of service; however, multicast capability is limited to a single logical LAN segment [HDBK92]. The lightweight profile provides real time data transfer

capability to various systems, as well as providing added options. The lightweight protocol suite consists of lightweight application services, the Xpress Transfer Protocol (XTP), ISO CLTP, ISO CLNP, ES/IS routing exchange protocol, IS/IS intra-domain routing protocol, LLC protocol and the FDDI protocols. This profile permits implementors to develop a set of communication services which support the performance and architecture requirements of a specific system. The lightweight profile provides a limited set of network management capabilities. If this service is required then the combined protocol suite must be utilized. Finally, while this lightweight protocol suite provides fast data transfer, it does not adhere to the ISO standard protocol and therefore is very system specific [KOCH91]. Figure 5 shows an overview of the lightweight profile.

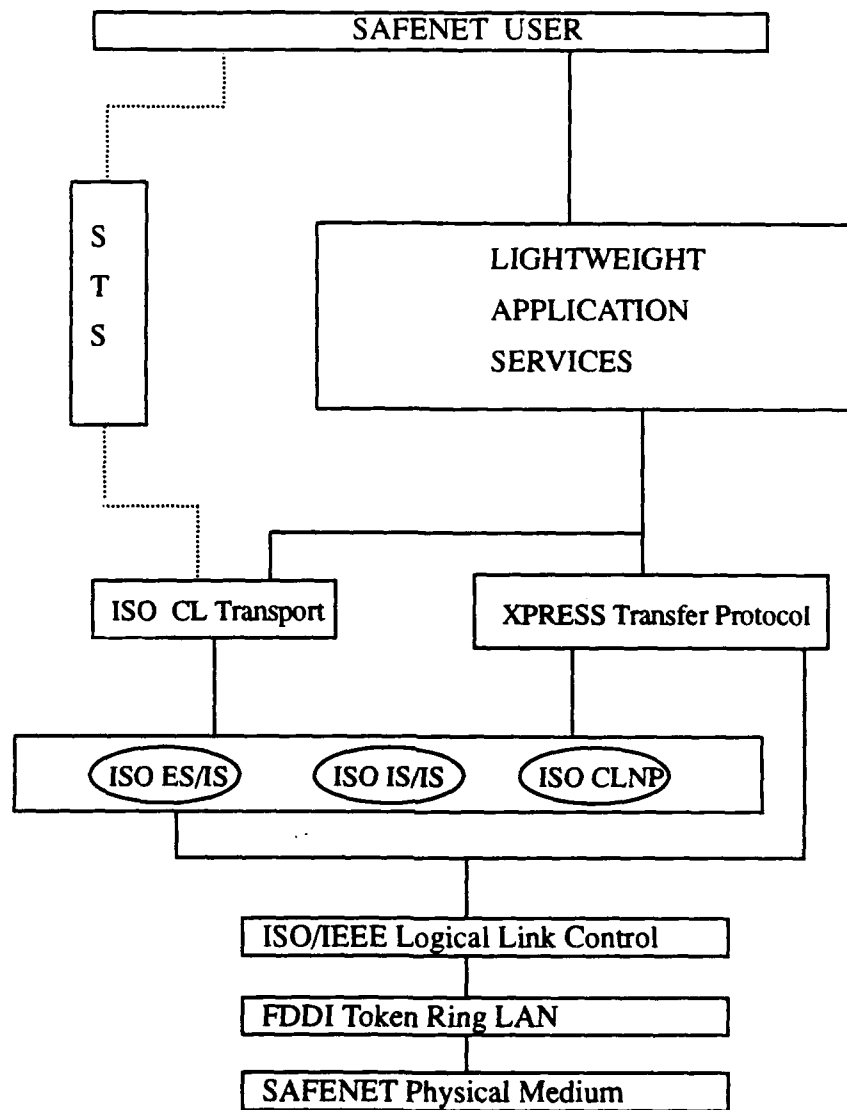


Figure 5 Overview of the Lightweight Profile

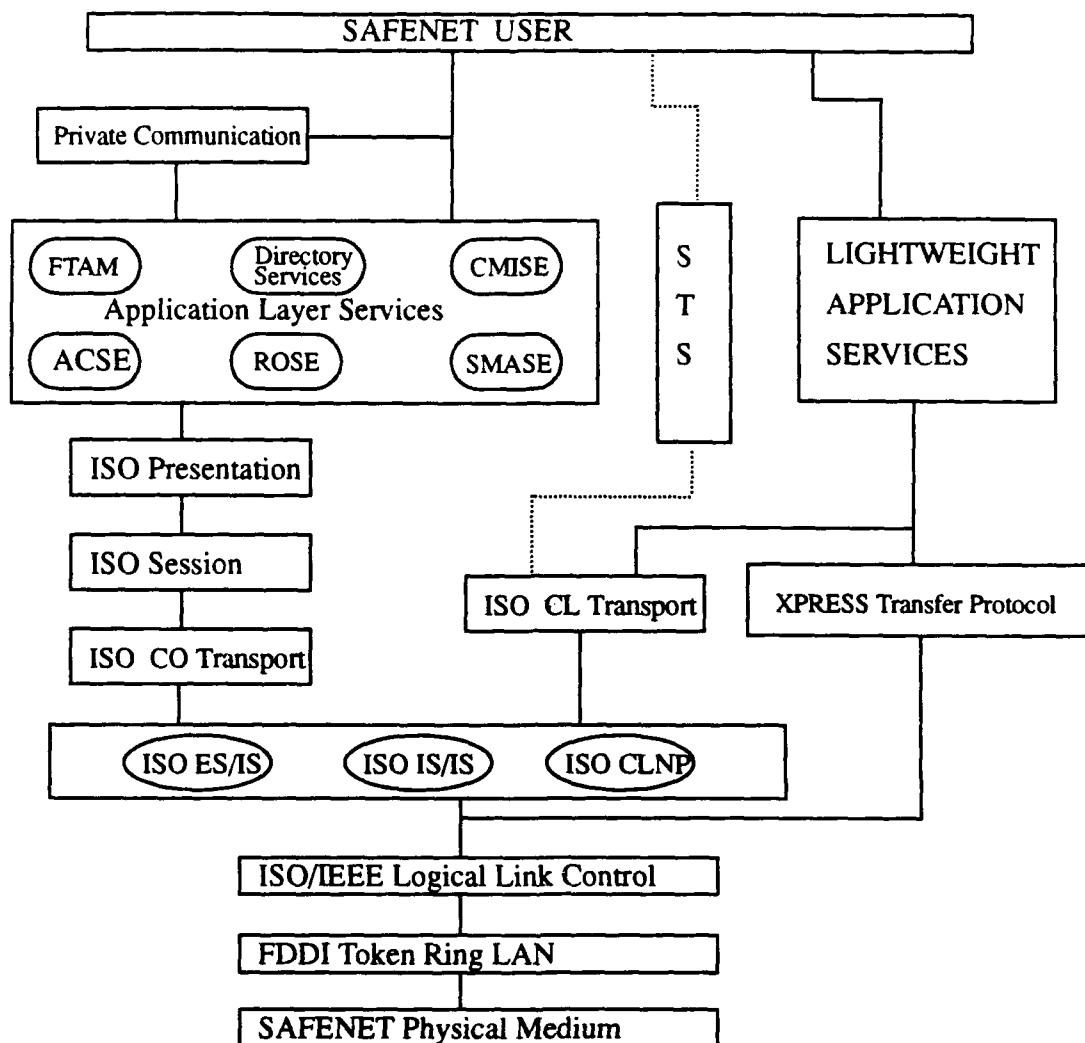


Figure 6 Overview of the Combined Profile

The combined profile is essentially the union of the OSI and lightweight protocol profiles. The combined protocol suite is intended for situations that require the capabilities of both the OSI and lightweight protocols. Therefore, all the protocols, services and capabilities of OSI and lightweight profiles are included. Additionally, network management protocols and services are provided for those protocol entities contained within the lightweight profile. However, because of the combined capability of this suite, it

requires much more complex development energy and cost [KOCH91]. Figure 6 shows an overview of the combined profile.

C. SAFENET TOPOLOGY

The basic topology design for SAFENET utilizes a redundant ring structure as shown in figure 1. The critical element of SAFENET's topology is the trunk Coupling Unit (TCU). The TCU device enables a station to insert or remove itself from a network ring. The TCU is a 2x2 optical bypass switch, which is controlled by an electrical signal from the attached station. The TCU has the capability to readily isolate a failed station from the network, thereby contributing to system reliability [PAIG90]. Optical signals are transmitted in opposite directions on each of the two rings. It is clear from this redundant ring topology that accurate and timely data flow is essential to the performance of SAFENET. Accordingly, as its name implies, SAFENET uses fiber optic technology as the physical medium in which data is exchanged. Consequently, this fiber optic technology forms the backbone of SAFENET's development.

D. SAFENET FIBER OPTIC DEVELOPMENT

The developers of SAFENET chose to employ a newer fiber optic technology over the older copper cables. For optical cables incorporate a number of excellent properties which provide data exchange for high-capacity transmission systems [JOHN87]. A major advantage of fiber optics is the large bandwidth times distance products obtained which allow data transmission rates of up to several Gbps [LI1983]. Furthermore, today's fibers typically offer bit rates of several hundred Mbps [IFOC84] and [LUND91]. Additionally, since glass is a dielectric medium, immune to electromagnetic interference and free from sparking, these optical fibers are useful in EMI-rich and other hostile environments. Low attenuation combined with its extremely small physical dimensions make optical fibers the physical medium of choice [FINL84].

As shown in Figure 1, each network ring is composed of TCUs and connecting trunk cables. The primary and secondary ring trunk cables are intended to be physically separated to enhance survivability in the event of battle damage. This placement strategy of allowing key network components (e.g., TCU and DAS) to be separated will allow the network to absorb some damage without the entire system losing its ability to operate.

It is understood that for an active ring either a node failure or a fiber break will cause a fatal crash. To correct this deficiency, SAFENET has added a second ring in the opposite direction as discussed earlier. This configuration allows for two types of network reconfiguration in the presence of a fault in the cable: ring hop/hold and ring wrap. Ring wrap is caused by a fault in the primary and secondary rings, the faulty sections of both rings are isolated by forming one or more rings out of the remaining portions of the primary and secondary rings [HDBK92].

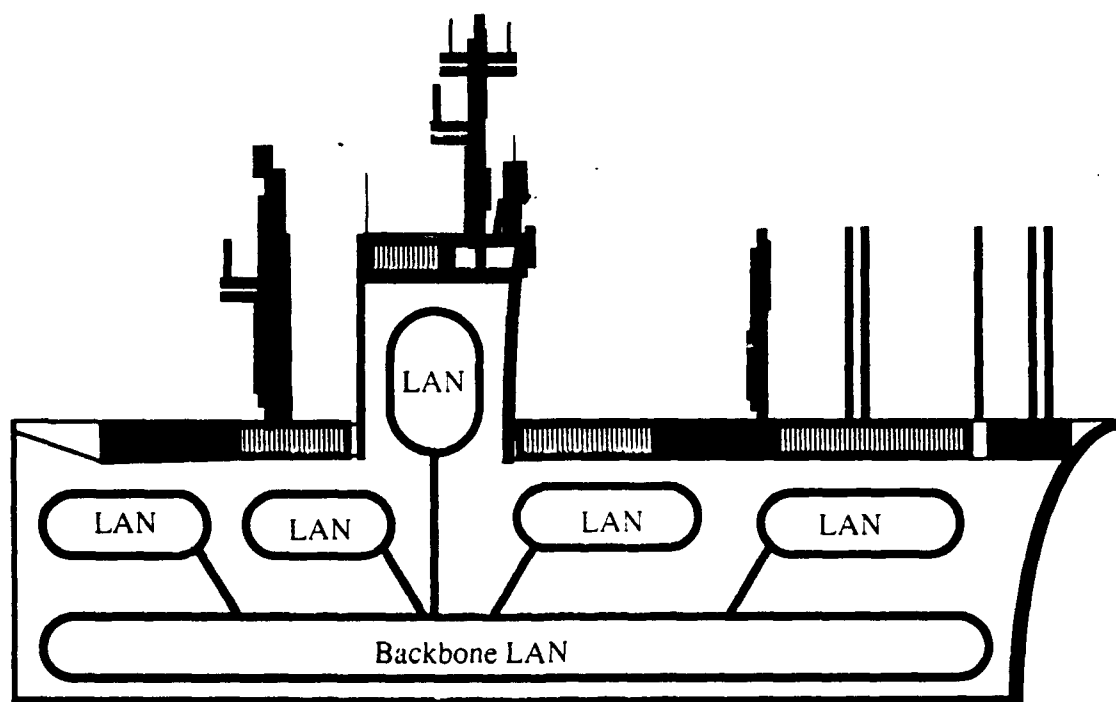


Figure 7 An overview of component systems comprising SAFENET

Figure 7 depicts the manner in which each component or warfare speciality area is unified into a whole. Each component, alone, is a system; yet, the synergism inherent in SAFENET's configuration and survivability features make it even more formidable.

IV. THE SAFENET STANDARD

A. THE SAFENET STANDARD AND ITS TESTABILITY

The SAFENET manual provides requirements for the implementation of fiber optic local area networks which are intended for use in support of mission critical computer resources. The SAFENET standard is organizationally well written, but it is large, and complex in its potential interprofile relationships. Each protocol within a SAFENET profile can be viewed as a finite state machine. The task of figuring out which protocol machines are expected to communicate within each profile can be awesome and daunting to some. While concrete design specifications are not expected to be contained in the manual, abstract design specifications should have and would have proven very useful to designers. As a result of neither abstract nor concrete design specifications, the SAFENET manual must be closely scrutinized and intra-profile relationships gleaned to garner all implicit relationships bit by bit in order to attain a formal specification.

The standards manual references in several cases existing standards; however, there are requirements which SAFENET references that are not yet completely formulated and are currently in draft status. SAFENET's Network Development Guidance and Conformance Test Guidance are two such requirements whose standards are listed as drafts. The Development Guidance and the Conformance Test Guidance are very crucial for SAFENET development. Systems Analyst and Designers will use these two manuals extensively to develop their implementation, in conjunction with the SAFENET standards manual. In addition, the Quality Assurance and Testing team will use these two manuals extensively to develop a test package. As a result of these two important publications being in their draft stages of development, the difficulty of testing a SAFENET implementation greatly increases. Consequently, the issue of whether we can implement SAFENET and perform the conformance testing without these two publications, begins to favor having the availability of both publications. Development and testability are greatly enhanced with the

availability of both the SAFENET Network Development Guidance and the SAFENET Conformance Test Guidance.

B. PROBLEMS WITH THE CURRENT SAFENET STANDARD

While SAFENET represents a major step forward in communications connectivity, this breakthrough is not without its share of problems and potential pitfalls. Typically, problems are associated with or attributed to any new introductory system. The current manual of SAFENET's standard is neither a users manual nor a technical manual. It is essentially a SAFENET standard document of specification that Development teams can use as guidelines for creating, implementing and testing a SAFENET network. Consequently, the current standard does not contain tutorials or a listing of descriptive features from a users' point of view. The standards manual is inadequate for describing specific features, user protocol interfaces, and for navigating between and within protocols. These features are expected to be contained in an implementor's users manual. Additionally, the current standard while containing much technical information is insufficient in that context, and the manual alone lacks the technical data necessary to facilitate a SAFENET installation in accordance with Military Specifications (MIL-SPECS).

Most of the fiber optic components used commercially meet MIL-SPECS, that is they conform to the quality control standard demanded by MIL-SPECS; however, the fiber optic bypass switch, contained within the TCU, did not perform in accordance with MIL-SPECS [GREE89]. As mentioned earlier, the fiber optic bypass switch bypasses a station which does not energize an electrical signal to this switch. Consequently, the SAFENET committee will have to expend research and development resources in this area to comply with MIL-SPECS. The specification of militarized components makes SAFENET lose the modular "plug" compatibility desired with standard off-the-shelf commercially available components, but this is seen as a necessary trade-off to enhance reliability and survivability.

In theory, ANSI FDDI can support the reliability and survivability features that are proposed for SAFENET, called dual path reconfiguration. SAFENET's approach to enhanced survivability was to initially specify the wrap method and after that specification is completed to actively work on development of a specification which uses both ring hop and wrap methods [GREE89]. The ANSI FDDI standard provides for the dual ring wrap reconfiguration technique; however, the ring hop reconfiguration technique is not supported in ANSI FDDI. The Navy's interest is expected to have a positive impact on development of the ring hop feature. But, whether this hop capability does become in reality a viable reconfiguration technique will depend on whether FDDI chip set manufacturers find the hop method cost effective to implement.

Another major area of concern is the support for time synchronization. The distribution of high precision, synchronized, digital time value among the components of a distributed real time system on a platform wide scale is one of the requirements that exist in all Navy SAFENET tactical systems. During the operational employment of a tactical system or platform, time synchronization is needed continually. The SAFENET Time Service (STS) provides for the distribution of time information and the synchronization of distributed clocks within a system. This capability is necessary for activities such as correlating information provided by various sensors to control weapons, to conduct post event reconstruction of critical events and to accomplish time critical diagnostic tests.

In searching for a candidate protocol for time synchronization, the SAFENET committee found that no true industry-wide standard existed [GREE89]. The Network Time Protocol (NTP), which utilizes a hardware clock of identifiable accuracy bound at each computing element employs a logical synchronized clock as its base. To support the STS each station or node is required to have a local time source, called a Network Clock. The Network Clock includes both the software and hardware components necessary to implement a time source. The STS is partitioned into three functional areas: the clock synchronization service, the user time service, and the time management service. Thus, the STS resident in each node uses the Network Clock to provide the current value of time,

clock quality and accuracy of time value to users in the local SAFENET node and to synchronize with other Network Clocks in the network. It is anticipated that major platforms will have clocks utilizing satellite synchronization; this concept appears to facilitate the Navy's application.

The prototyping activity in progress within the NGCR community is looked upon to supply the critical answer as to whether distributed Network Clocks will satisfy naval requirements. Surprisingly, SAFENET has established no error bounds or performance requirements for the individual Network Clocks in a network. It is assumed that the system implementors will use components of sufficient quality to meet the system specific requirements. The quality of the components used will have a direct impact on the level of synchronization achieved. In all aspects, this represents a potential pitfall; clock synchronization performance is a critical area that will require close attention during system design. The performance requirements of the system will need to be specified and the appropriate configuration parameters determined.

One primary necessity for naval tactical systems is the need to support real time communications, be it, periodic and aperiodic, multicast and point to point, or low latency communications. The ability to delay or even abort low value communications in favor of more urgent communications is a problem for both present and future applications, particularly in the event of momentary insufficient data communication resources [GREE89]. Because of the nature of timed token behavior, ANSI FDDI provides only two levels for data frames, synchronous and asynchronous traffic. This methodology has no effect on which station obtains the right to transmit data next. The basis for SAFENET I, the IEEE 802.5 standard provides a three bit priority field which supports eight levels of priority and is used for selecting the token holder, the next station allowed to transmit frames. The viewpoint of the SAFENET committee is that eight levels of priority is inadequate for meeting system requirements. SAFENET did not provide any real time specification for this area and thus this led to the SAFENET Lightweight Protocol profile. The Lightweight Protocol candidate selected was the XTP protocol which is non-

proprietary, has the potential for industry standardization and provides the needed support for real time communications. The development of SAFENET requirements used the XTP protocol as a reference for what is possible and practical and also as a means of feedback to the XTP developers the requirements which were not met by the current version of the XTP protocol. Thus, the key and potential problem for the Lightweight Protocol profile is how well it supports real time, prioritized traffic in present and future applications.

Presently, in the Lightweight profile only the XTP SAFENET hex address format can be used [HDBK92]. Particularly in stressful moments, hex addressing does not lend itself very well to human recall as the primary means of station addressing. Within the OSI profile, given a logical name known by the application, the Directory Services will provide the needed addressing information. However, no specification exists for real time users not using OSI as the communication protocol, and since SAFENET allows a non-standard approach for the Application, Presentation, and Session layers as well as the Lightweight Protocol underneath, a major problem exists here.

The Lightweight profile's multicast capability envisioned for SAFENET is limited to a single logical LAN segment; this may not be the optimal approach. Naval combat platforms in hostile environments should have the ability to broadcast across multiple LANs to report tactical and strategic casualties (Engineering, Weapons, etc.) to decision makers. The issue of multicast capability to multiple LANs should be reviewed to determine its value to decision makers.

SAFENET addresses the issue of clock granularity, but in terms of clock accuracy, no guidance is established for the performance requirements of the individual network clocks in a network. Determining the system clock resolution is needed to accurately determine the amount of time taken in point to point data transfers. For example, consider a radar target that is passed to weapons for engagement. Clock resolution and synchronization is what is required in this real time scenario for a successful engagement. Without these two, resolution and synchronization, how can a target be engaged, if we do not know whether the data is late, therefore useless or an actual prediction. This issue of clock performance,

resolution and synchronization is a potential pitfall because the determination of what is "sufficient" component quality is left up to the system engineer.

Security as pertains to the protection of data that is transferred on the network is the final area of concern. Although it may not be necessary for all implementations of SAFENET to provide security services, security implementations will be necessary in some platforms. For example, a platform with an implementation that involves data transfers of multiple classification will require a risk analysis to determine which security services to provide. Satisfying the security requirements as provided in MIL-HDBK-818-1 (still in drafting stage) may prove difficult to implement and yet, still conform to SAFENET's standard.

V. TESTING FIBER OPTIC LINKS

The design points for a communication link are many and require careful consideration. The band width-length specified for SAFENET Laser or LED with multimode, graded index fiber can support data rates from 10 to several hundred Mbps over distances ranging from 10 meters to 200 kilometers. To design a reliable SAFENET communications data link, a thorough survey and analysis of system requirements is necessary.

The maximum tolerable bit error rate for the system must be determined. The bit error rate is the probability that an error has been detected in a received bit. The determination of the bit error rate is a critical element in the total SAFENET communications system performance. The bit error rate for a metallic connection is approximately 10^{-6} ; whereas, a bit error rate of 10^{-9} is commonly used for fiber optic data links. The bit error rate is also a product of receiver sensitivity. The total allowable power loss for the link is the difference in the power provided by the source and the power required by the detector. Additionally, some spare power must be present to account for temperature variations, diode aging and bend loss on the fiber. The above criteria represent the major points in fiber optic data link design.

In the case of ANSI FDDI, the single most important factor is the bit error probability (P_e). If the P_e is too high, the need for frame retransmission occurs more frequently; if it is too small, the system will be prohibitively expensive. The P_e for FDDI is 2.5×10^{-10} which is easily attainable with current optical fiber technology. As long as the signal-to-noise ratio is sufficient, the required P_e is attainable. Conversely, if the signal-to-noise ratio is insufficient, the P_e will tend toward certainty (1).

A. LOSS BUDGET

A loss budget analysis is important for ensuring that the SAFENET system will meet or exceed performance limits. In conventional radio frequency systems like Ethernet or Token Ring, the signal-to-noise ratio must be large enough to support a specified P_e . For an optical system, the goal is the same, but the calculations are based on losses specific to the optical net. The ANSI FDDI standard specifies a P_e of less than 2.5×10^{-10} [ANNA88]. Robert Kimball provides a detailed explanation of the different losses associated with FDDI [KIMB89]. The reason for conducting this analysis is to determine whether or not the installation will meet the FDDI requirements and thus be in conformance with SAFENET. The general form of the decision statistic is:

$$P \geq \mu_l + \mu_d + \mu_m + 2\sigma_T$$

P represents the available power, defined as 11 dB for FDDI. The first term on the right hand side of the inequality, μ_l , is the sum of the aggregate component losses in the link. These losses include propagation losses due to irregularities in the fiber, connector losses, splice losses, higher order mode losses (due to refraction inside the fiber), and the Media Interface Connector (MIC) ferrule delta. The MIC ferrule delta accounts for the difference between the precision test ferrule and a production ferrule [KIMB89]. MIC is the plug and receptacle pair that makes the physical connection between the optical fibers and the transmitter or receiver.

The second term on the right hand side, μ_d , is the dispersion penalty, which accounts for dispersion losses in the optical fiber. This is a function of bit rate and of several chromatic characteristics of the LEDs used in FDDI. Within the dispersion penalty equation, the average segment length component accounts for links that consist of several spliced segments. This accounts for the bandwidth concatenation phenomena, which may

cause a bandwidth increase in concatenated fibers over what is normally expected in a single, unbroken fiber.

The third term on the right hand side, μ_m , is the system margin. It represents a catch-all that allows for variations in the cable plant and a factor that compensates for timing variations between the light level at the output of the fiber and the light received at the lens on the receiver.

The final term on the right hand side, $2\sigma_T$, is the total variance of the link loss distribution and is defined as a function of the variances of the dispersion penalty and the loss distribution.

The final step is to substitute all the intermediate results for the right hand side terms back into the original equation to verify that we have not exceeded the loss budget. If the right hand side of the equation exceeds 11 dB, one would need to go back to the SAFENET installation and figure out where the loss budget can be improved. The area that would provide the greatest change with the least effort would be the aggregate component loss factor. Two ways to improve this factor would be to shorten the links between transmitter and receiver or reduce the number of connectors. Obviously, there will be instances where it is impractical to alter this component; consequently, other components on the right hand side of the inequality will have to be evaluated.

B. SYSTEM THROUGHPUT

Theoretically, networks can approach 100% transmission efficiency, but there are certain trade-offs to be addressed. Contention based protocols which approach 100% transmission efficiency have excessive wait delays associated with them. Collision free protocols such as ANSI FDDI are better suited for approaching the transmission efficiency limit.

1. Clock Accuracy Verification

Timing analysis is critical to determining how well the SAFENET system performs over the network. Recent studies have shown that bottlenecks (choke points) in the protocol stacks and the processors are more detrimental to network speed than the raw data transfer rate. In order to determine how well the SAFENET profiles perform, it is necessary to time different data transfers and compare them. Determining the system clock resolution is needed to accurately determine how long data transfers from one point to another take. Inaccurate timing would jeopardize the validity of any data collected.

2. Timing Test Procedure

To attain a more meaningful result from data transfer analyses, data transfers should be conducted under normal net loading and under no load "ideal" conditions. Test sets should consist of two groups of file transfers. The test files should be selected based on size. The criteria for size is outlined in the following paragraphs.

The largest data set must be large enough to exceed the size of the buffers on the interface cards. Care must be taken in selecting an appropriate size file and yet minimize the effects (by percentage) of overhead.

The next file has to be smaller than the aforementioned file, but larger than the size of an FDDI packet. The space reserved for data is 4478 bytes in an FDDI packet. Once more, care must be taken to select an appropriate size file while minimizing the effects (by percentage) of overhead. FDDI has no minimum packet size. Percentage of overhead is calculated by dividing the number of bytes of overhead by the number of data bytes, then multiplying the result by 100.

Validating operational specifications set forth by manufacturers and SAFENET's standard and testing for proper installation are dominant reasons for collecting system measurements. Of particular interest are the fiber attenuation, band width, bit error rate, transmitter and receiver coupled power, connector loss, splice loss and the signal-to-noise ratio. Some measurements for conformance determination are to be taken before, during

and after installation. Fortunately, test equipment in conjunction with the theoretical and empirical analyses are available to measure and test each area of concern. The Consultative Committee International Telegraph and Telephone (CCITT) has produced recommendations for test methods to which, it is hoped, most SAFENET component manufacturers and test equipment manufacturers will adhere. The testing procedure is, test all parts and components before installation, test all parts and components after installation, and perform integration tests by testing subsystems and entire systems after installation. For complete systems tests, ensure data test sets emulate valid data; this will ensure that the results obtained are meaningful.

VI. TESTING SAFENET'S IMPLEMENTATION PROFILES

A. TEST METHOD

In this section, the test procedure is described; the following description is actually a refinement of the method described in [MILL90]. From the SAFENET standard, Finite State Machine Diagrams of the protocols contained within the SAFENET profiles were created. From these diagrams, predicate-action tables for systems of communicating machines were created for the OSI and Light Weight implementations. The test procedure's initial input is a protocol specified as a system of communicating machines and the output is a complete test sequence and an Input/Output diagram. In order to proceed from the specification of a protocol machine or profile implementation to a test sequence, identification of the shared and local variables is necessary. The shared and local variables present a way for the tester to provide input to and observe output from the machine during testing. The test of each edge, transition, is treated as a separate, individual test, and may modify some or all of the shared and local variables.

The format of each single edge, transition sequence is

$$S_I i_1, i_2, \dots, i_n / o_1, o_2, \dots, o_m S_E$$

where S_I is the state of the machine at the start of the test, i_1, i_2, \dots, i_n are the values of the input variables before the test, o_1, o_2, \dots, o_m are the values of the output variables after the test, and S_E is the state of the machine at the end of the test. The input and output variables are determined before testing begins and are taken from the shared and local variables of the machine or profile. The procedure consists of preliminary steps, the test sequence generating procedure, and refining steps.

1. Preliminary Steps

1. From the machine specification finite state diagram, mark each transition whose name appears on more than one transition or edge. Each such instance for a given name is given a separate distinguishing label.

2. From the predicate-action table note the number of distinct conditions or clauses in each enabling predicate. Mark each clause.

3. For each shared variable x , determine if x is an input variable, output variable or both an input and output variable. For each x which is both input and output, split x into two variables, x_I and x_O for testing purposes.

4. For each local variable l , determine if l is used as an interface to the higher layer user of this profile or protocol. If such is the case, mark l as input, output or both. Each such local variable is an input variable if it appears in an enabling predicate and a output variable if it appears in an action on the left side of an assignment arrow. If l is both input and output, it is split into variables l_I and l_O for testing purposes.

Step 1 is to ensure that each instance of each transition is tested. A protocol specification may have the same transition name on more than one arc; this step provides a degree of certainty that every arc is tested. Step 2 ensures that each clause is tested individually, if possible. An enabling predicate may consist of several clauses, any one of which may be true, allowing the transition to execute. In steps 3 and 4, the shared and local variables are identified. Shared variables provide the means of communication between the machine and other protocol entities within a profile. Local variables allow communication with the user of the protocol or profile. In essence, these variables are the means the test designer has of giving inputs to the machine and observing its actions.

In some system of communicating machine specifications, additional variables are defined that are used internally by the protocol or profile machine and are not visible to the user (upper layer(s) of the protocol) or the tester. Typically, such variables are counters or

array indices. These variables should not appear in the test sequence as they are not under the direct control or observation of the tester.

2. Test Sequence Generating Procedure

1. $S_I \leftarrow \text{initial_state}$

2. Let $t = (p,a)$ be an untested transition from an arbitrary state. What this notation means is that the current transition being tested, t , has the predicate, p , as input and the action, a , as output.

(a) Determine the values of the input variables which make exactly one of the untested values of p true. Check to see if these values allow any other transition from this state to be executed. If so, set additional input variables to values that ensure that only the transition being tested is enabled. Fill in the necessary input variables, and mark the others DC for "don't care."

(b) Determine and mark the expected values for the output variables. In addition record the expected values assumed by the local variables.

(c) Determine the expected next state and set S_E to it.

(d) Determine if S_E is transient; if it is not, label it as a "stop state" and proceed to step 3. Within the confines of the test procedure, a state is transient if one or more of its enabling predicates is true upon reaching the state. This means that the machine can proceed to another state without having to wait for further input from the tester.

(e) Attempt to make S_E into a stop state by setting DC values such that when state S_E is reached, none of its enabling predicates are true. If successful, proceed to step 3.

(f) S_E is a transient state. If more than one transition leaving S_E is enabled, arbitrarily select one and set the input not yet specified, such that only one transition leaving S_E is enabled. Set $t = (p,a)$ to this transition.

3. Output this test $S_I i_1, i_2, \dots, i_n / o_1, o_2, \dots, o_m S_E$ as the next test in the sequence.

4. Mark the clause just tested. If all clauses in transition t are now tested, mark t as tested. If all transitions are now marked as tested, exit to "refining steps." Otherwise, proceed to step 5.

5. Set S_I to S_E . If S_I is a stop state, proceed to step 2.; otherwise, proceed to step 2(b).

Step 2(a) attempts to test each clause separately. For well designed protocols this is generally true. It is vital in that the tester knows which transition was enabled, and as a result, caused the transition to occur. In the event that it is not possible to individually test each clause, the test designer must set the input variables such that the clauses are tested as meticulously as possible. It is quite possible that such clauses may be tested in conjunction with one another. However, if a clause cannot be tested individually, the question of clause necessity to the specification arises.

Steps 2(d), 2(e) and 2(f) are concerned with transient states. If execution of a state immediately enables another transition, it may prove difficult or even impossible for the tester to verify that the values contained in the output variables are as expected. For such a circumstance, the transient state and possible multiple enabling transitions that can not be controlled with these test methods, could indicate the need to modify the specification for better testability.

Step 5 initializes the start state of the next test in the sequence to the ending state of the current test. The advantage here is that the ordering of the tests follow the order of their occurrence in the actual profile implementation. In order to exercise all parts of a protocol or profile implementation, some transitions may have to be executed more than once. In such a circumstance, judgement by the test designer may be needed. This is not necessarily a cause for concern; in the normal operation of a profile or protocol machine, certain transitions may be executed more than others. Consequently, during testing the same trend will likely be true.

3. Refining Steps

1. Construct the Input/Output state diagram from the test sequence. In this step, the stop state information is also used, assuming that there is at least one stop state.

2. For each state, determine a shortest Unique Input/Output (UIO) sequence (if one exists). Append the UIO sequence for S_E to the end of the test sequence. If no UIO sequence for the current S_E exists, continue testing transactions (extending the sequence) until an S_E with a UIO is visited.

3. Check for converging transactions which are difficult to test and may require special handling. These transactions are marked as potential problems.

In step 1, the Input/Output diagram is constructed from the test sequence and is a tool to help the test designer ensure completeness. This diagram is constructed directly from the test sequence with the knowledge of "stop states." The diagram's initial state will be initial state S_I ; additional states are added for each stop state is encountered. The arcs are directed, and labeled with the with the values of the input and output variables.

The I/O diagram generated from the test sequence can be viewed as an incomplete finite state machine specification. However, there is a relationship to the specification machine, because there is a tour through the specified transactions. It is not identical to the specification; there are states which are transitory in the specification and does not appear in the IO diagram.

The purpose of the final UIO sequence in step 2 serves to verify that the last state which was reached in the test sequence is indeed the current state of the protocol or profile machine. Because the details of the machine's implementation are assumed to be "hidden" from the tester, the existence of at least state with a UIO sequence is necessary. Without the UIO sequence, there is no way of knowing if the last transition tested, left the machine in the expected state.

In actuality, the converging transitions, identified in step 3, are distinct transactions, with identical labels, which originate from different states but terminate at the same state.

The existence of converging conditions can not be eliminated always and, therefore, complicates the role of the test designer and makes error detection difficult. These circumstances may naturally occur in the specification of a protocol, but should be marked for special observation.

B. APPLICATION OF TEST METHOD

In this section the test generating procedure is illustrated using derived specifications for two of the SAFENET standard profiles: OSI profile and Lightweight profile. The profiles are first specified as a system of communicating machines and then the procedure is given.

1. OSI Profile Specification

The specification of the OSI profile consists of the predicate-action table (Table 1), the specification for each protocol within the profile, given in Figure 8, and the inter-process shared variable **MEDIUM**, shown in Figure 9. The single intra-process shared variable **Transfer** is used to model the network node's internal bus, which is shared by the protocols within a node or station. An internal transmission is modeled by a write into this shared variable. The first field "**Transfer.T**" takes the value T or F, which is used to indicate whether the frame represents a time synchronization frame or a data frame. The second field, **DA** for Destination Address, contains the address of the protocol machine to which the message is transmitted. The next field, **SA** for Source Address contains the originator's address. Fields four through eleven contain the values T or F and are used to control the intra-process routing; based on the values contained in these variables, the frame's Destination Address is determined. Finally, the data field contains the data block itself. The single shared variable **MEDIUM** is used to model the bus, which is shared by each machine or network node. A transmission onto the bus is modeled by a write into this shared variable. The first field "**MEDIUM.T**" takes the value T or F, which is used to indicate whether the frame represents a time synchronization frame or a data frame. The second

field, DA for Destination Address, contains the address of the network station to which the message is transmitted. The next field, SA for Source Address contains the originator's address; finally, the data field contains the data block itself.

The OSI profile is defined by a finite state machine, a set of local variables and a predicate-action table. The initial state of each profile machine is state 0, and the shared variables MEDIUM and Transfer are initially set to contain the respective address of one of the stations or protocol machines in the DA field.

The local variables inbuf, outbuf, etc. are used for storing data blocks to be transmitted to or received from other protocols. Outbuf is an array, and can store a potentially large number of data blocks.

The initial state of each profile machine is state 0 and all local variables are initially set to empty. The inter-process shared variable MEDIUM initially contains the address of one station in its DA field. Therefore, the initial system state tuple is (0,...,0) and the first transition taken by a station holding the token will be xmit_time, or xmit_msg.

Each profile machine has 18 states. In the initial state, 0, the station is quiescent, merely waiting for an incoming message, a transmit message signal, or a transmit time synchronization signal. If a frame appears in the variable MEDIUM with the network node's own address, the transition to state 1 is taken. When taking the xmit_time transition, in state 2, the station transmits the data blocks that it has via Transfer, moving to state 3. In state 3, the station transmits the data blocks it has, moving to state 6. As specified by the SAFENET standard, synchronization frames are sent via the ISO CLNP Protocol [HDBK92] page. 37. In state 6, the data blocks are formed into datagrams and transmitted, moving to state 17. In state 17, the station transmits any data blocks it has moving to state 18. In state 18, the station transmits until its token holding time expires or all of its messages are sent; the station then returns to state 0. When taking the xmit_clear_logical_msg transition, in state 8, the station transmits the blocks that it has, moving to either state 9, state 10 or state 11. If in state 9, the station transmits the data blocks it has moving to state 14. If in state 10, the station transmits the data blocks it has

moving to state 14. If in state 11, the station transmits the data blocks it has moving to state 12. From state 12, the station transmits the data blocks it has moving to state 13. If in state 13, the station transmits the data blocks it has moving to state 14. If in state 14, the station transmits the data blocks it has moving to state 15. From state 15, the station moves to station 16 after transmitting its data blocks. When in state 16, the station transmits the data blocks it has moving to states 4, 5, or 6; this transition is based on the message size and its destination address' location. If in state 4, the station transmits the data blocks it has moving to state 17; from states 5, or 6, the station transmits its data blocks, moving to state 17. In state 17, the station transmits any data blocks it has moving to state 18. In state 18, the station transmits until its token holding time expires or all its messages are sent; the station then returns to state 0.

The receiving profile station, like all other stations not in possession of the token, will be in state 0. The message will appear in MEDIUM, with the receiving station's address in the DA field. The receive transition to state 1 will be taken. In state 1 the data block is copied and the MEDIUM is cleared by the ready transition. By clearing the MEDIUM, the receiving station enables the sending station to return to its initial state (0) or its sending state (18).

For this simplified high-level specification, the channels inter-process and intra-process are assumed to be error free. This means that the clearing of the medium by the receiver can be taken as an acknowledgment by the sender. Hence, there is no requirement for timers, time-outs or error checking on the channel. Although some of the finer details of the OSI profile are omitted, this specification contains the main idea of the OSI profile, and provides sufficient detail for the generation of a test sequence.

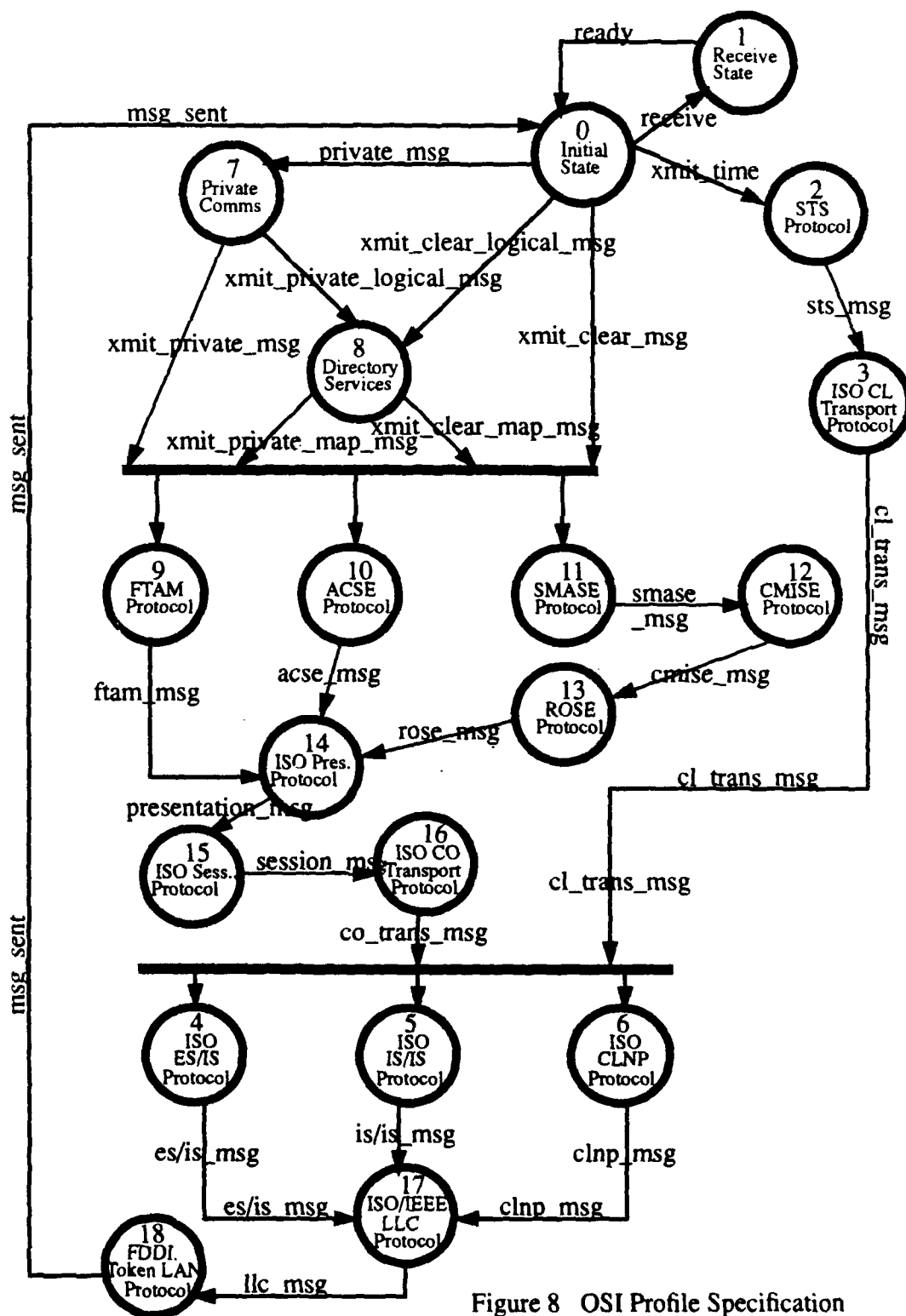


Figure 8 OSI Profile Specification

Each machine within the OSI profile in Figure 8 performs the following:

- State 0 In the initial state, the machine is quiescent, merely waiting to process a request or transmission.
- State 1 In the receive state, the machine copies an incoming message from the bus and acknowledges receipt of the message by clearing the bus.
- State 2 The SAFENET Time Service protocol provides for the distribution of time information and the synchronization of distributed clocks within a system.
- State 3 In addition to Lightweight and Xpress Transfer Protocol support, the OSI Connectionless transport protocol directly supports STS's protocol data unit transfer. It provides the user with the ability to transmit a single unit of data, datagram, without the requirement of a connection being established.
- State 4 The ISO End System-to-Intermediate System routing exchange protocol passes address information among all stations that are on the same LAN segment or through intermediate stations. The ES/IS protocol provides stations with the ability to associate a data link layer address with a given network layer address.
- State 5 The ISO Intermediate System-to-Intermediate System intra-domain routing protocol provides SAFENET networks with dynamic determination of routes used to pass data between intermediate systems.
- State 6 The ISO Connectionless Network protocol provides services for network routing and for the segmentation and reassembly of transport layer messages that are too large for the underlying communications service. Additionally, the ISO CLNP has multicast data transfer capability, but limits the scope of transfers to users on a single LAN segment.
- State 7 The Private Communications protocol provides a means for secure communications between network stations.
- State 8 The Directory Services provides a mapping from "user friendly" names (application entity titles) to presentation service access point addresses required in communication instances.
- State 9 The File Transfer, Access, and Management protocol provides services for transferring information in the form of files among application processes and file stores.
- State 10 The Association Control Service Element protocol provides services for the establishment and termination of application layer associations and a standard service for application layer protocols to communicate common parameters.
- State 11 The System Management Application Service Element protocols specifies the management functions which are supported in a system node, and defines the semantics and abstract syntaxes of information transferred. It uses CMISE for communication.
- State 12 The Common Management Information Service Element protocol provides a common message framework for management procedures supplying both data-manipulation and notification/operation-oriented services.
- State 13 The Remote Operations Service Element protocol is used in support of CMISE, and provides a simple, uniform service for remotely invoking operations and

then receiving correlated replies to those operations.

- State 14 The ISO Presentation protocol is concerned with the syntax of data the data exchanged between application entities and resolves differences in format and data representation. The presentation layer defines the syntax used between application entities and provides for the selection and subsequent modification of the representation to be used.
- State 15 The ISO Session protocol provides the mechanism for controlling the dialogue between applications. At a minimum, it provides a means for two application processes to establish and use a connection.
- State 16 The ISO Connection-Oriented Transport protocol provides for the establishment, maintenance, and termination of a logical connection between transport users. It allows connection-related features such as flow control, error control, and sequenced delivery.
- State 17 The Logical Link Control protocol provides three services: Unacknowledged connectionless service which supports point-to-point, multipoint and broadcast in a datagram style of service, Connection-oriented services which provides flow control, sequencing, and error recovery, Acknowledged connectionless service which provides for acknowledgment of individual frames and supports point-to-point transfers.
- State 18 The FDDI Token LAN protocol provides the ability to get data on and off the physical medium in a controlled manner.

t = time; m = map; p = private; DA = destination address; SA = source address

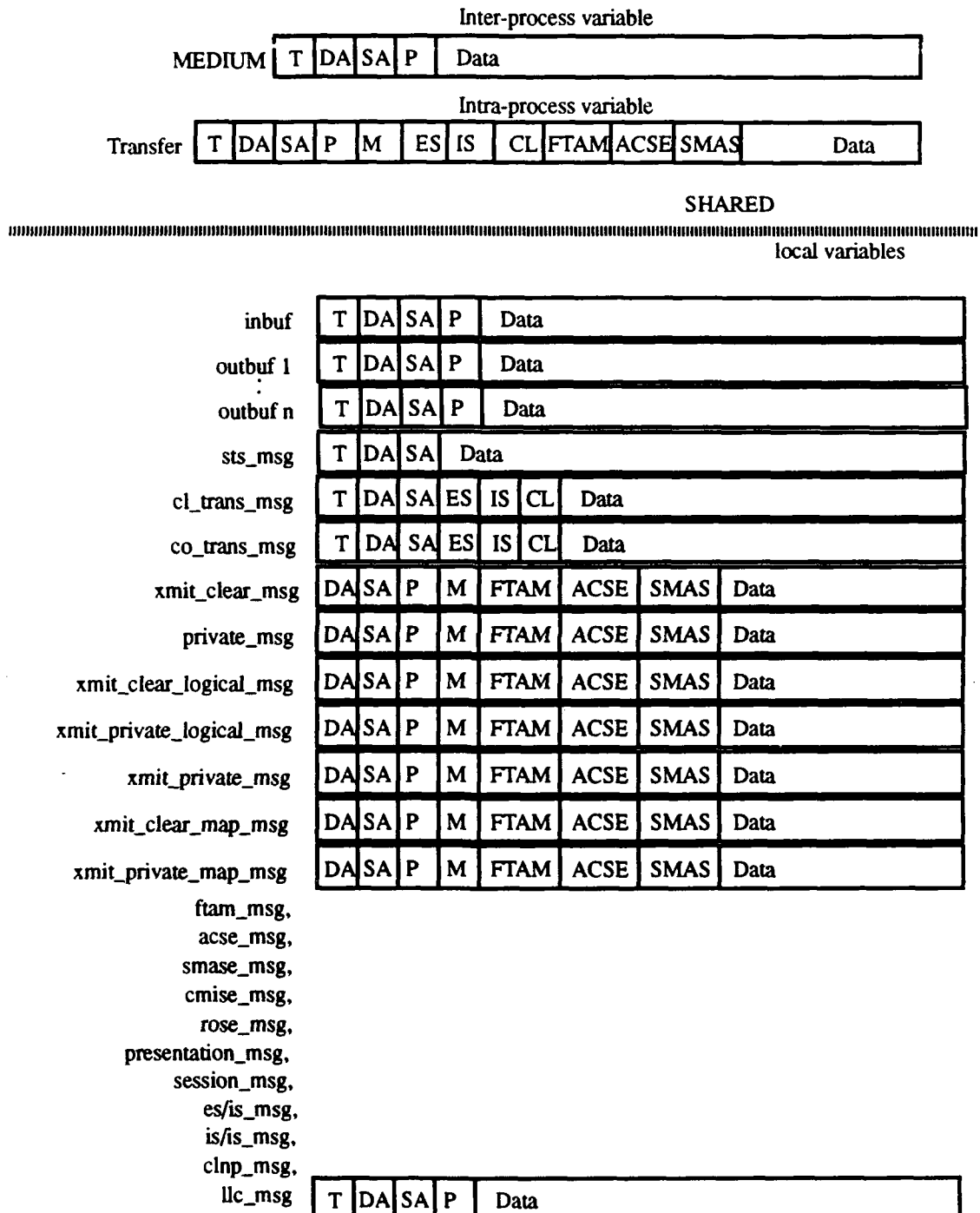


Figure 9 OSI Network Node Data Structure

Table 1: OSI PREDICATE-ACTIONS

Transition	Predicate	Action
receive	MEDIUM.DA = i	inbuf \leftarrow MEDIUM
ready	true	MEDIUM $\leftarrow \emptyset$
xmit_time	outbuf(n) $\neq \emptyset \wedge$ outbuf(n).t = true	Transfer \leftarrow outbuf(n) \wedge outbuf(n) $\leftarrow \emptyset$
sts_msg	sts_msg(t, data) = (true, msg)	Transfer \leftarrow sts_msg \wedge sts_msg $\leftarrow \emptyset$
private_msg(p_msg)	p_msg(p, m, data) = (true, DC, msg)	Transfer \leftarrow p_msg \wedge p_msg $\leftarrow \emptyset$
xmit_clear_logical_msg	xclm(p, m, data) = (false, true, msg)	Transfer \leftarrow xclm \wedge xclm $\leftarrow \emptyset$
xmit_private_logical_msg	xplm(p, m, data) = (true, true, msg)	Transfer \leftarrow xplm \wedge xplm $\leftarrow \emptyset$
xmit_clear_msg(xcm)	xcm(p, m, smase) = (false, false, true)	Transfer \leftarrow xcm \wedge xcm $\leftarrow \emptyset$
xmit_clear_msg(xcm)	xcm(p, m, acse) = (false, false, true)	Transfer \leftarrow xcm \wedge xcm $\leftarrow \emptyset$
xmit_clear_msg(xcm)	xcm(p, m, ftam) = (false, false, true)	Transfer \leftarrow xcm \wedge xcm $\leftarrow \emptyset$
xmit_clear_map_msg	xcmm(p, m, ftam) = (false, true, true)	Transfer \leftarrow xcmm \wedge xcmm $\leftarrow \emptyset$
xmit_clear_map_msg	xcmm(p, m, smase) = (false, true, true)	Transfer \leftarrow xcmm \wedge xcmm $\leftarrow \emptyset$
xmit_clear_map_msg	xcmm(p, m, acse) = (false, true, true)	Transfer \leftarrow xcmm \wedge xcmm $\leftarrow \emptyset$
xmit_private_map_msg	xpmm(p, m, smase) = (true, true, true)	Transfer \leftarrow xpmm \wedge xpmm $\leftarrow \emptyset$
xmit_private_map_msg	xpmm(p, m, ftam) = (true, true, true)	Transfer \leftarrow xpmm \wedge xpmm $\leftarrow \emptyset$
xmit_private_map_msg	xpmm(p, m, acse) = (true, true, true)	Transfer \leftarrow xpmm \wedge xpmm $\leftarrow \emptyset$
xmit_private_msg(xpm)	xpm(p, m, ftam) = (true, false, true)	Transfer \leftarrow xpm \wedge xpm $\leftarrow \emptyset$
xmit_private_msg(xpm)	xpm(p, m, acse) = (true, false, true)	Transfer \leftarrow xpm \wedge xpm $\leftarrow \emptyset$
xmit_private_msg(xpm)	xpm(p, m, smase) = (true, false, true)	Transfer \leftarrow xpm \wedge xpm $\leftarrow \emptyset$
ftam_msg	ftam_msg $\neq \emptyset$	Transfer \leftarrow ftam_msg \wedge ftam_msg $\leftarrow \emptyset$
acse_msg	acse_msg $\neq \emptyset$	Transfer \leftarrow acse_msg \wedge acse_msg $\leftarrow \emptyset$
smase_msg	smase_msg $\neq \emptyset$	Transfer \leftarrow smase_msg \wedge smase_msg $\leftarrow \emptyset$
cmise_msg	cmise_msg $\neq \emptyset$	Transfer \leftarrow cmise_msg \wedge cmise_msg $\leftarrow \emptyset$
rose_msg	rose_msg $\neq \emptyset$	Transfer \leftarrow rose_msg \wedge rose_msg $\leftarrow \emptyset$
presentation_msg(pm)	pm $\neq \emptyset$	Transfer \leftarrow pm \wedge pm $\leftarrow \emptyset$
session_msg(sm)	sm $\neq \emptyset$	Transfer \leftarrow sm \wedge sm $\leftarrow \emptyset$
co_trans_msg(cotm)	cotm $\neq \emptyset \wedge$ cotm.es = true	Transfer \leftarrow cotm \wedge cotm $\leftarrow \emptyset$
co_trans_msg(cotm)	cotm $\neq \emptyset \wedge$ cotm.is = true	Transfer \leftarrow cotm \wedge cotm $\leftarrow \emptyset$
co_trans_msg(cotm)	cotm $\neq \emptyset \wedge$ cotm.cl = true	Transfer \leftarrow cotm \wedge cotm $\leftarrow \emptyset$
cl_trans_msg(cltm)	cltm $\neq \emptyset$	Transfer \leftarrow cltm \wedge cltm $\leftarrow \emptyset$
es/is_msg	es/is_msg $\neq \emptyset$	Transfer \leftarrow es/is_msg \wedge es/is_msg $\leftarrow \emptyset$
is/is_msg	is/is_msg $\neq \emptyset$	Transfer \leftarrow is/is_msg \wedge is/is_msg $\leftarrow \emptyset$
clnp_msg	clnp_msg $\neq \emptyset$	Transfer \leftarrow clnp_msg \wedge clnp_msg $\leftarrow \emptyset$
llc_msg	llc_msg $\neq \emptyset$	Transfer \leftarrow llc_msg \wedge llc_msg $\leftarrow \emptyset$
msg_sent	true	MEDIUM $\leftarrow \emptyset$

1. OSI Test Sequence Generation

First the preliminary steps are carried out; these steps determine the exact format of the tests. The measures employed are primarily concerned with input and output variables. After the preliminary steps, the tests are generated. For ease of reference the numbering below corresponds to the steps in the test procedure.

a. Preliminary Steps

(1) From Figure 8's Lightweight profile specification finite state diagram, we see that all transition labels are unique; therefore, no action is required.

(2) All transitions have single clause enabling predicates; therefore, no action is required.

(3) The shared variable MEDIUM is both an input and an output variable; therefore it is split into two variables MEDIUM_I and MEDIUM_O for testing purposes. The intra-process shared variable Transfer is both an input and an output variable; therefore it is split into two variables Transfer_I and Transfer_O for testing purposes

(4) The local variables outbuf, sts_msg, private_msg, xmit_private_logical_msg, xmit_clear_logical_msg, xmit_clear_msg, xmit_clear_map_msg, xmit_private_map_msg, xmit_private_msg, ftam_msg, acse_msg, smase_msg, cmise_msg, rose_msg, presentation_msg, session_msg, co_trans_msg, cl_trans_msg, llc_msg, clnp_msg, es/is_msg and is/is_msg are both input and output variables; therefore they are split into two respective variables, for example private_msg_I and private_msg_O, for testing purposes.

Note that in step 2, the co_trans_msg and xmit_time are not separated into two different clauses because both conditions must be true for the transition to be enabled.

From these preliminary steps, we can see that the test will adhere to the following form:

$S_I \text{ MEDIUM}_I \text{ Transfer}_I \text{ outbuf}_I \dots \text{llc_msg}_I / \text{MEDIUM}_O \text{ Transfer}_O \text{ outbuf}_O \dots \text{llc_msg}_O \text{ inbuf } S_E$

Now we are ready to begin generating the test sequence.

b. Test Sequence Generation

(1) We begin in the initial state, 0. In step 2 we may choose any untested transition emanating from state 0; we select the `xmit_clear_msg` transition.(step 9).

2(a) According to the predicate-action table, to enable this transition the local variable `xmit_clear_msg` must contain data for processing and the DA field of `xmit_msg` is assumed to be state 9's address. The remaining fields may have any values, and are indicated by an "x" in Table 2. The other input variables are set to DC for "don't care."

2(b) When the transition occurs, Transfer copies the data from `xmit_clear_msg`, and `xmit_clear_msg` is set to empty.

2(c) S_E is set to the expected end state for this test, which is state 9.

(3) Noting that the next state is a stop state, this completes the first test in the sequence, and the appropriate values are shown in Table 2.

(4) This clause and transition are now marked "tested."

(5) The value of S_I is now set to 9 and another iteration starting at step10 is called for.

The next iteration of the procedure selects the `ftam_msg` transition, and the values selected are shown as the tenth test entered in Table 2. The expected ending state for this tenth test is 14. The next iteration of the procedure selects the `presentation_msg` transition, and the values selected are shown as the eleventh test entered in Table 2. The expected ending state for this tenth test is 15. From state 15 in test step 12, we take the `session_msg` transition. The expected ending state resulting from this transition is 16.

At the next iteration, the `co_trans_msg` transition is taken; the expected ending state for this thirteenth test is 4. From state 4, we take the `es/is_msg` transition. In test step fourteen, the expected ending state resulting from this transition is 17. From state 17, we take the `llc_msg` transition; the expected ending state for this fifteenth test is 18. From state

18, we exercise the `msg_sent` transition using the “true” predicate, which leads back to the initial state.

The remaining untested transitions are executed in a similar manner resulting in a final test sequence of 356 steps. The values of the input and output variables for all of these tests are shown in the appendix.

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts_ msg _i	private_ msg _i	xmit_priv ate_logica l_msg _i	xmit_clear_ logical_ msg _i	xmit_clear_ msg _i	xmit_map_ clear_ msg _i	xmit_ private_ map_msg _i	xmit_pr ivate_ msg _i
1 receive	0	x.i.x.x	DC	DC DC								
2 ready	1	DC	DC	DC DC								
3 xmit_time	0	DC	DC	TDA...								
4 sts_msg	2	DC	DC	DC DC	TDA,x,x							
5 cl_trans_msg	3	DC	DC	DC DC								
6 clnp_msg	6	DC	DC	DC DC								
7 llc_msg	17	DC	DC	DC DC								
8 msg_sent	18	DC	DC	DC DC								
9 xmit_clear_msg	0	DC	DC	DC DC				DA,x,FF,FF,x				
10 fltam_msg	9	DC	DC	DC DC								
11 presentation_msg	14	DC	DC	DC DC								
12 session_msg	15	DC	DC	DC DC								
13 co_trans_msg	16	DC	DC	DC DC								
14 es/is_msg	4	DC	DC	DC DC								
15 llc_msg	17	DC	DC	DC DC								
16 msg_sent	18	DC	DC	DC DC								
17 xmit_clear_msg	0	DC	DC	DC DC				DA,x,FF,FF,x				
18 fltam_msg	9	DC	DC	DC DC								
19 presentation_msg	14	DC	DC	DC DC								
20 session_msg	15	DC	DC	DC DC								
21 co_trans_msg	16	DC	DC	DC DC								
22 is/is_msg	5	DC	DC	DC DC								
23 llc_msg	17	DC	DC	DC DC								
24 msg_sent	18	DC	DC	DC DC								
25 xmit_clear_msg	0	DC	DC	DC DC				DA,x,FF,FF,x				
26 fltam_msg	9	DC	DC	DC DC								
27 presentation_msg	14	DC	DC	DC DC								
28 session_msg	15	DC	DC	DC DC								
29 co_trans_msg	16	DC	DC	DC DC								
30 is/is_msg	6	DC	DC	DC DC								
31 llc_msg	17	DC	DC	DC DC								
32 msg_sent	18	DC	DC	DC DC								

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msg _i	acse_ msg _i	smase_ msg _i	cmise_ msg _i	rose_ msg _i	presen- tation_ msg _i	session_ msg _i	co trans_ msg _i	cl_trans_ msg _i	es/is_ msg _i	is/is_ msg _i	clnp_ msg _i	llc_ msg _i
1												
2												
3												
4												
5								T.DA.x.FFT				
6											T.DA.x.x.x	
7												T.DA.x.x.x
8												
9												
10 x.DA.x.x.x												
11					x.DA.x.x.x							
12						x.DA.x.x.x						
13							x.DA.x.TFF.x					
14									x.DA.x.x.x			
15												x.DA.x.x.x
16												
17												
18 x.DA.x.x.x												
19					x.DA.x.x.x							
20						x.DA.x.x.x						
21							x.DA.x.FTF.x					
22										x.DA.x.x.x		
23												x.DA.x.x.x
24												
25												
26 x.DA.x.x.x												
27					x.DA.x.x.x							
28						x.DA.x.x.x						
29							x.DA.x.FTF.x					
30											x.DA.x.x.x	
31												x.DA.x.x.x
32												

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbufo (1,2)	sts_ msg ₀	private_ msg ₀	xmit_priv ate_logical_ msg ₀	xmit_clear_ msg ₀	xmit_clear_ map_msg ₀	xmit_ private_ map_msg ₀	xmit_priv ivate_ msg ₀	flam_ msg ₀
1 DC	DC	DC DC								
2 Ø	DC	DC DC								
3 DC	TxSAxx	Ø DC								
4 DC	TxSAx	DC DC	Ø							
5 DC	TxSAFFTx	DC DC								
6 DC	TxSAxx	DC DC								
7 DC	TxSAxx	DC DC								
8 Ø	DC	DC DC								
9 DC	xxSAFFTx	DC DC				Ø				
10 DC	xxSAxx	DC DC								Ø
11 DC	xxSAxx	DC DC								
12 DC	xxSAxx	DC DC								
13 DC	xxSAFFTx	DC DC								
14 DC	xxSAxx	DC DC								
15 DC	xxSAxx	DC DC								
16 Ø	DC	DC DC								
17 DC	xxSAFFTx	DC DC				Ø				Ø
18 DC	xxSAxx	DC DC								
19 DC	xxSAxx	DC DC								
20 DC	xxSAxx	DC DC								
21 DC	xxSAFFTx	DC DC								
22 DC	xxSAxx	DC DC								
23 DC	xxSAxx	DC DC								
24 Ø	DC	DC DC								
25 DC	xxSAFFTx	DC DC				Ø				Ø
26 DC	xxSAxx	DC DC								
27 DC	xxSAxx	DC DC								
28 DC	xxSAxx	DC DC								
29 DC	xxSAFFTx	DC DC								
30 DC	xxSAxx	DC DC								
31 DC	xxSAxx	DC DC								
32 Ø	DC	DC DC								

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msg0	smase msg0	cmise msg0	rose msg0	presen tation msg0	session msg0	co trans msg0	cl trans msg0	es/is msg0	is/is msg0	clnp msg0	llc_ msg0	inbuf	Se
1												x.x.SA.x.x	1
2													0
3													2
4													3
5													6
6							Ø						17
7										Ø			18
8													0
9													9
10													14
11				Ø									15
12					Ø								16
13						Ø							4
14								Ø					17
15											Ø		18
16													0
17													9
18													14
19				Ø									15
20					Ø								16
21						Ø							5
22									Ø				17
23											Ø		15
24													0
25													9
26													14
27				Ø									15
28					Ø								16
29						Ø							6
30										Ø			17
31											Ø		18
32													0

2. Lightweight Profile Specification

The specification of the Lightweight profile consists of the predicate-action table (Table 2), the specification for each protocol within the profile, given in Figure 10, and the inter-process shared variable MEDIUM, shown in Figure 11. The single intra-process shared variable Transfer is used to model the network node's internal bus, which is shared by the protocols within a node or station. An internal transmission is modeled by a write into this shared variable. The first field Transfer.T takes the value T or F, which is used to indicate whether the frame represents a time synchronization frame or a data frame. The second field, DA for Destination Address, contains the address of the protocol machine to which the message is transmitted. The next field, SA for Source Address contains the originator's address. Fields four through eleven contain the values T or F and are used to control the intra-process routing; based on the values contained in these variables, the frame's Destination Address is determined. Finally, the data field contains the data block itself. The single shared variable MEDIUM is used to model the bus, which is shared by each machine or network node. A transmission onto the bus is modeled by a write into this shared variable. The first field MEDIUM.T takes the value T or F, which is used to indicate whether the frame represents a time synchronization frame or a data frame. The second field, DA for Destination Address, contains the address of the network station to which the message is transmitted. The next field, SA for Source Address contains the originator's address; finally, the data field contains the data block itself.

The Lightweight profile is defined by a finite state machine, a set of local variables and a predicate-action table. The initial state of each profile machine is state 0, and the shared variables MEDIUM and Transfer are initially set to contain the respective address of one of the stations or protocol machines in the DA field.

The local variables inbuf, outbuf, etc. are used for storing data blocks to be transmitted to or received from other protocols. Outbuf is an array, and can store a potentially large number of data blocks.

The initial state of each profile machine is state 0 and all local variables are initially set to empty. The inter-process shared variable MEDIUM initially contains the address of one station in its DA field. Therefore, the initial system state tuple is (0,...,0) and the first transition taken by a station holding the token will be xmit_time, or xmit_msg.

Each profile machine has 10 states. In the initial state, 0, the station is quiescent, merely waiting for an incoming message, a transmit message signal, or a transmit time synchronization signal. If a frame appears in the variable MEDIUM with the network node's own address, the transition to state 1 is taken. When taking the xmit_time transition, in state 2, the station transmits the data blocks that it has via Transfer, moving to state 3. In state 3, the station transmits the data blocks it has, moving to state 8. As specified by the SAFENET standard synchronization frames are sent via the ISO CLNP Protocol [HDBK92] page. 37. In state 8, the data blocks are formed into datagrams and transmitted, moving to state 9. In state 9, the station transmits any data blocks it has moving to state 10. In state 10, the station transmits until its token holding time expires or all its messages are sent; the station then returns to state 0. When taking the xmit_msg transition, in state 4, the station transmits the blocks that it has, moving to either state 3 or state 5. If in state 3, the station transmits the data blocks it has moving to states 6, 7, or 8; this transition is based on the message size and its destination address' location. If in state 5, the station transmits the data blocks it has moving to states 6, 7, 8, or 9; From states 5, 6, 7, or 8, the station transmits its data blocks, moving to state 9. In state 9, the station transmits any data blocks it has moving to state 10. In state 10, the station transmits until its token holding time expires or all of its messages are sent; the station then returns to state 0.

The receiving profile station, like all other stations not in possession of the token, will be in state 0. The message will appear in MEDIUM, with the receiving station's address in the DA field. The receive transition to state 1 will be taken. In state 1 the data block is copied and the MEDIUM is cleared by the ready transition. By clearing the MEDIUM, the receiving station enables the sending station to return to its initial state (0) or its sending state (10).

For this simplified high-level specification, the channels inter-process and intra-process are assumed to be error free. This means that the clearing of the medium by the receiver can be taken as an acknowledgment by the sender. Hence, there is no requirement for timers, time-outs or error checking on the channel. Although some of the finer details of the Lightweight profile are omitted, this specification contains the main idea of the Lightweight profile, and provides sufficient detail for the generation of a test sequence.

Table 3: LIGHTWEIGHT PREDICATE-ACTIONS

Transition	Predicate	Action
receive	MEDIUM.DA = i	inbuf \leftarrow MEDIUM
ready	true	MEDIUM $\leftarrow \emptyset$
xmit_time	outbuf(n) $\neq \emptyset \wedge$ outbuf(n) t = true	Transfer \leftarrow outbuf(n) \wedge outbuf(n) $\leftarrow \emptyset$
sts_msg	sts_msg(t, data) = (true, msg)	Transfer \leftarrow sts_msg
xmit_msg	xmit_msg $\neq \emptyset$	Transfer \leftarrow xmit_msg \wedge xmit_msg $\leftarrow \emptyset$
lightwt_cl_msg(lwcm)	lwcm $\neq \emptyset$	Transfer \leftarrow lwcm \wedge lwcm $\leftarrow \emptyset$
xfer_xtp_msg(xxm)	xxm $\neq \emptyset$	Transfer \leftarrow xxm \wedge xxm $\leftarrow \emptyset$
cl_msg(xpm)	xpm(t, cl) = (true, true)	Transfer \leftarrow xpm \wedge xpm $\leftarrow \emptyset$
xtp_rte_msg(xrm)	_xrm(t, es) = (true, true)	Transfer \leftarrow xrm \wedge xrm $\leftarrow \emptyset$
xtp_rte_msg(xrm)	xrm(t, is) = (true, true)	Transfer \leftarrow xrm \wedge xrm $\leftarrow \emptyset$
xtp_rte_msg(xrm)	xrm(t, cl) = (true, true)	Transfer' \leftarrow xrm \wedge xrm $\leftarrow \emptyset$
es/is_msg	es/is_msg $\neq \emptyset$	Transfer \leftarrow es/is_msg \wedge es/is_msg $\leftarrow \emptyset$
is/is_msg	is/is_msg $\neq \emptyset$	Transfer \leftarrow is/is_msg \wedge is/is_msg $\leftarrow \emptyset$
clnp_msg	clnp_msg $\neq \emptyset$	Transfer \leftarrow clnp_msg \wedge clnp_msg $\leftarrow \emptyset$
xtp_msg	xtp_msg $\neq \emptyset$	Transfer \leftarrow xtp_msg \wedge xtp_msg $\leftarrow \emptyset$
llc_msg	llc_msg $\neq \emptyset$	Transfer \leftarrow llc_msg \wedge llc_msg $\leftarrow \emptyset$
msg_sent	true	MEDIUM $\leftarrow \emptyset$

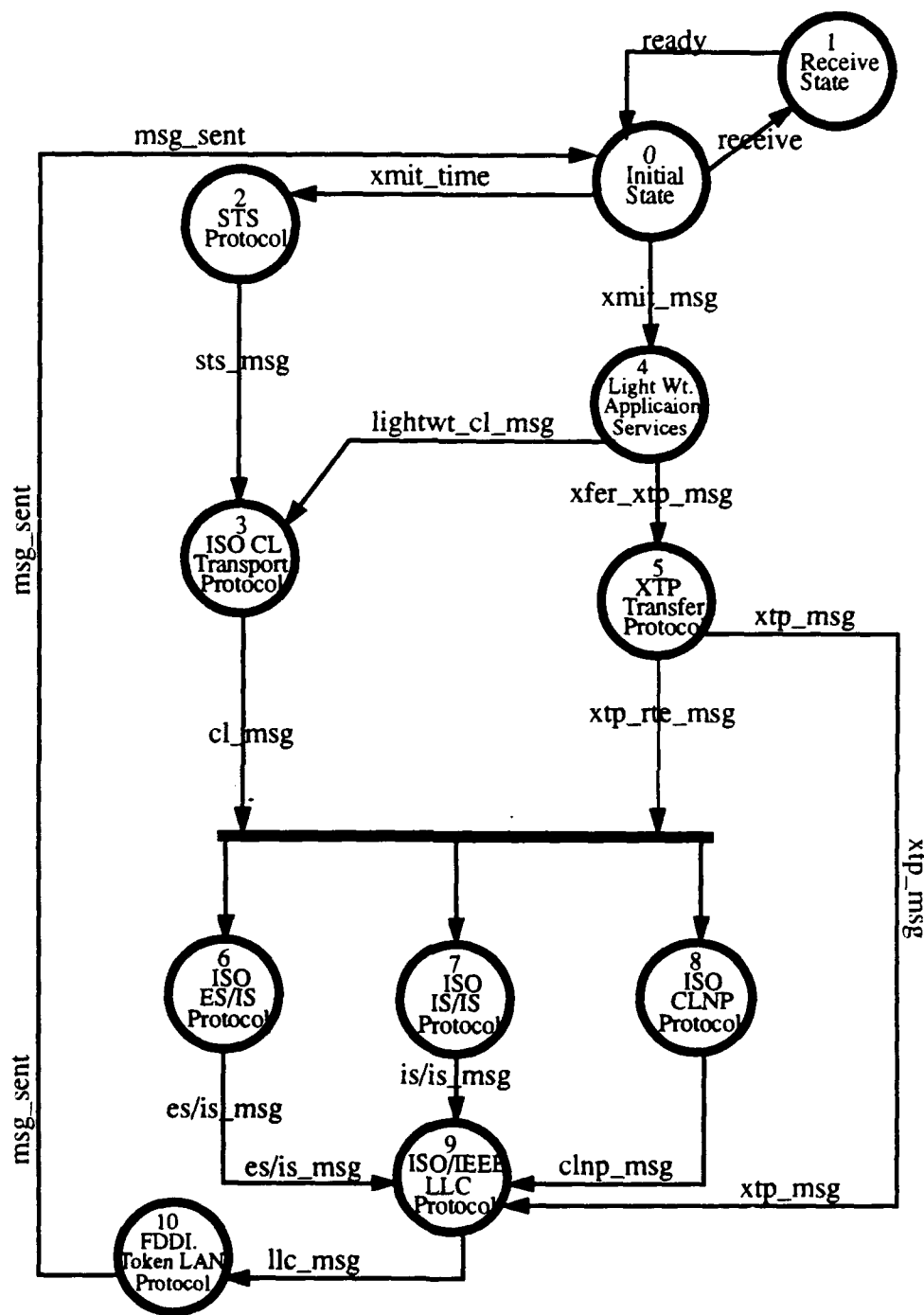


Figure 10 Lightweight Profile Specification

Each machine within the Lightweight profile in Figure 10 performs the following:

- State 0 In the initial state, the machine is quiescent, merely waiting to process a request or transmission.
- State 1 In the receive state, the machine copies an incoming message from the bus and acknowledges receipt of the message by clearing the bus.
- State 2 The SAFENET Time Service protocol provides for the distribution of time information and the synchronization of distributed clocks within a system.
- State 3 In addition to Lightweight and Xpress Transfer Protocol support, the OSI Connectionless transport protocol directly supports STS's protocol data unit transfer. It provides the user with the ability to transmit a single unit of data, datagram, without the requirement of a connection being established.
- State 4 The Lightweight application services consist of a set of communication service primitives which can be implemented to provide a user with direct, efficient data transfer capabilities.
- State 5 The Xpress Transfer protocol provides services which achieve increased efficiency and performance by combining the connection process with the data transfer process.
- State 6 The ISO End System-to-Intermediate System routing exchange protocol passes address information among all stations that are on the same LAN segment or through intermediate stations. The ES/IS protocol provides stations with the ability to associate a data link layer address with a given network layer address.
- State 7 The ISO Intermediate System-to-Intermediate System intra-domain routing protocol provides SAFENET networks with dynamic determination of routes used to pass data between intermediate systems.
- State 8 The ISO Connectionless Network protocol provides services for network routing and for the segmentation and reassembly of transport layer messages that are too large for the underlying communications service. Additionally, the ISO CLNP has multicast data transfer capability, but limits the scope of transfers to users on a single LAN segment.
- State 9 The Logical Link Control protocol provides three services: Unacknowledged connectionless service which supports point-to-point, multipoint and broadcast in a datagram style of service, Connection-oriented services which provides flow control, sequencing, and error recovery, Acknowledged connectionless service which provides for acknowledgment of individual frames and supports point-to-point transfers.
- State 10 The FDDI Token LAN protocol provides the ability to get data on and off the physical medium in a controlled manner.

t = time; DA = destination address; SA = source address

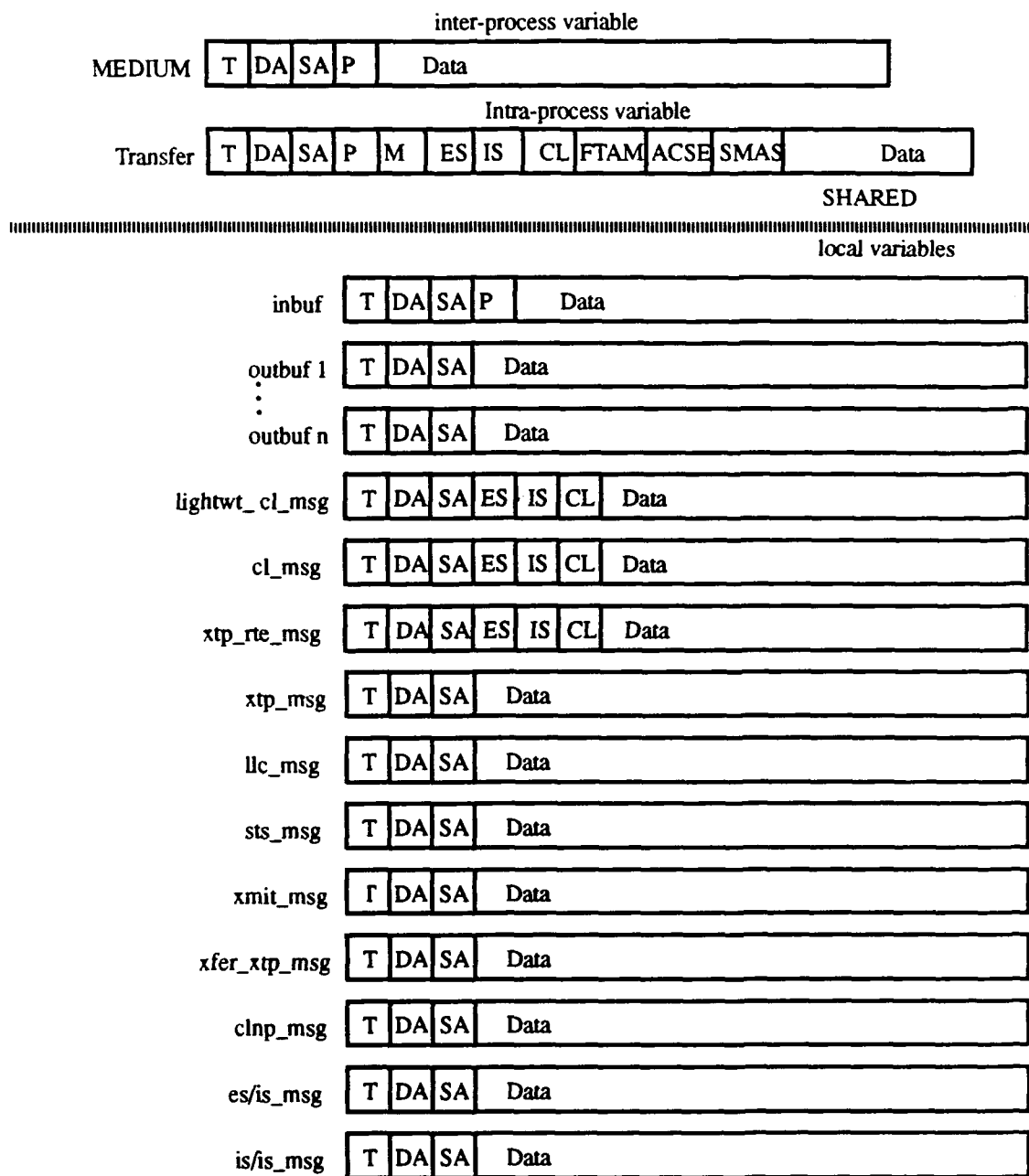


Figure 11 Lightweight Network Nodes Data Structure

3. Lightweight Test Sequence Generation

First the preliminary steps are carried out; these steps determine the exact format of the tests. The measures employed are primarily concerned with input and output variables. After the preliminary steps, the tests are generated. For ease of reference the numbering below corresponds to the steps in the test procedure.

a. Preliminary Steps

(1) From Figure 10's Lightweight profile specification finite state diagram, we see that all transition labels are unique; therefore, no action is required.

(2) All transitions have single clause enabling predicates; therefore, no action is required.

(3) The shared variable MEDIUM is both an input and an output variable; therefore it is split into two variables MEDIUM_I and MEDIUM_O for testing purposes. The intra-process shared variable Transfer is both an input and an output variable; therefore it is split into two variables Transfer_I and Transfer_O for testing purposes

(4) The local variables outbuf, sts_msg, lightwt_cl_msg, cl_msg, xtp_rte_msg, xtp_msg, llc_msg, xmit_msg, xfer_xtp_msg, clnp_msg, es/is_msg and is/is_msg are both input and output variables; therefore they are split into two respective variables, for example xmit_msg_I and xmit_msg_O, for testing purposes.

Note that in step 2, the xtp_rte_msg and xmit_time are not separated into two different clauses because both conditions must be true for the transition to be enabled.

From these preliminary steps, we can see that the test will adhere to the following form:

$S_I \text{ MEDIUM}_I \text{ Transfer}_I \text{ outbuf}_I \dots \text{llc_msg}_I / \text{MEDIUM}_O \text{ Transfer}_O \text{ outbuf}_O \dots \text{llc_msg}_O \text{ inbuf } S_E$

Now we are ready to begin generating the test sequence.

b. Test Sequence Generation

(1) We begin in the initial state, 0. In step 2 we may choose any untested transition emanating from state 0; we select the `xmit_msg` transition.

2(a) According to the predicate-action table, to enable this transition the local variable `xmit_msg` must contain data for processing and the DA field of `xmit_msg` is assumed to be state 4's address. The remaining fields may have any values, and are indicated by an "x" in Table 4. The other input variables are set to DC for "don't care."

2(b) When the transition occurs, Transfer copies the data from `xmit_msg`, and `xmit_msg` is set to empty.

2(c) S_E is set to the expected end state for this test, which is state 4.

(3) Noting that the next state is a stop state, this completes the first test in the sequence, and the appropriate values are shown in Table 4.

(4) This clause and transition are now marked "tested."

(5) The value of S_I is now set to 4 and another iteration starting at step 4 is called for.

The next iteration of the procedure arbitrarily selects the `lightwt_cl_msg` transition, and the values selected are shown as the fourth test entered in Table 4. The expected ending state for this fourth test is 3.

At the next iteration, the `cl_msg` transition is taken; the expected ending state for this fifth test is 8. From state 8, we take the `clnp_msg` transition. The expected ending state resulting from this transition is 9. From state 9, we take the `llc_msg` transition; the expected ending state for this seventh test is 10. From state 10, we exercise the `msg_sent` transition using the "true" predicate, which leads back to the initial state.

The remaining untested transitions are executed in a similar manner resulting in a final test sequence of 32 steps. The values of the input and output variables for all of these tests are shown in Table 4.

Table 4: TEST SEQUENCE FOR THE LIGHTWEIGHT PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts_ msg _i	xmit_ msg _i	lightwt_ cl_ msg _i	xfer_ xtp_ msg _i	cl_ msg _i	xtp_ rte_ msg _i	es/is_ msg _i	is/is_ msg _i
1 receive	0	x,x,x,x,x	DC	DC DC								
2 ready	1	DC	DC	DC DC								
3 xmit_msg	0	DC	DC	DC DC		x,DA,x,x						
4 lightwt_cl_msg	4	DC	DC	DC DC			x,DA,x,FFTx					
5 cl_msg	3	DC	DC	DC DC					x,DA,x,FFTx			
6 clnp_msg	8	DC	DC	DC DC								
7 llc_msg	9	DC	DC	DC DC								
8 msg_sent	10	DC	DC	DC DC								
9 xmit_msg	0	DC	DC	DC DC		x,DA,x,x						
10 xfer_xtp_msg	4	DC	DC	DC DC				x,DA,x,x				
11 xtp_rte_msg	5	DC	DC	DC DC						x,DA,x,TFEx		
12 es/is_msg	6	DC	DC	DC DC							x,DA,x,x	
13 llc_msg	9	DC	DC	DC DC								
14 msg_sent	10	DC	DC	DC DC								
15 xmit_msg	0	DC	DC	DC DC		x,DA,x,x						
16 xfer_xtp_msg	4	DC	DC	DC DC				x,DA,x,x				
17 xtp_rte_msg	5	DC	DC	DC DC						x,DA,x,FFTx		
18 is/is_msg	7	DC	DC	DC DC								x,DA,x,x
19 llc_msg	9	DC	DC	DC DC								
20 msg_sent	10	DC	DC	DC DC								
21 xmit_msg	0	DC	DC	DC DC		x,DA,x,x						
22 xfer_xtp_msg	4	DC	DC	DC DC				x,DA,x,x				
23 xtp_rte_msg	5	DC	DC	DC DC						x,DA,x,FFTx		
24 clnp_msg	8	DC	DC	DC DC								
25 llc_msg	9	DC	DC	DC DC								
26 msg_sent	10	DC	DC	DC DC								
27 xmit_time	0	DC	DC	T,DA,...								
28 sts_msg	2	DC	DC	DC DC	T,DA,x,x							
29 cl_msg	3	DC	DC	DC DC					x,DA,x,FFTx			
30 clnp_msg	8	DC	DC	DC DC								
31 llc_msg	9	DC	DC	DC DC								
32 msg_sent	10	DC	DC	DC DC								

Table 4: TEST SEQUENCE FOR THE LIGHTWEIGHT PROFILE (CONTINUED)

clnp_msg1	clnp_msg1	llc_msg1	MEDIUM ₀	Transfero	outbufo (1,2)	sts_msgo	xmit msgo	lightwt cl_msgo	xfer xtp_msgo	cl_msgo	xtp - rie msgo
1			DC		DC DC						
2			Ø		DC DC						
3			DC	x.x.SA.x	DC DC		Ø				
4			DC	x.x.SAFFTx	DC DC			Ø			
5			DC	x.x.SAFFTx	DC DC					Ø	
6	x.DA.x.x		DC	x.x.SA.x	DC DC						
7		x.DA.x.x	DC	x.x.SA.x	DC DC						
8			Ø		DC DC						
9			DC	x.x.SA.x	DC DC		Ø				
10			DC	x.x.SA.x	DC DC				Ø		
11			DC	x.x.SA.TFFx	DC DC						Ø
12			DC	x.x.SA.x	DC DC						
13		x.DA.x.x	DC	x.x.SA.x	DC DC						
14			Ø		DC DC						
15			DC	x.x.SA.x	DC DC		Ø				
16			DC	x.x.SA.x	DC DC				Ø		
17			DC	x.x.SA.FTx	DC DC						Ø
18			DC	x.x.SA.x	DC DC						
19		x.DA.x.x	DC	x.x.SA.x	DC DC						
20			Ø		DC DC						
21			DC	x.x.SA.x	DC DC		Ø				
22			DC	x.x.SA.x	DC DC				Ø		
23			DC	x.x.SAFFTx	DC DC						Ø
24	x.DA.x.x		DC	x.x.SA.x	DC DC						
25		x.DA.x.x	DC	x.x.SA.x	DC DC						
26			Ø		DC DC						
27			DC	x.x.SA.x	Ø DC						
28			DC	T.x.SA.x	DC DC	Ø					
29			DC	x.x.SAFFTx	DC DC					Ø	
30	x.DA.x.x		DC	x.x.SA.x	DC DC						
31		x.DA.x.x	DC	x.x.SA.x	DC DC						
32			Ø		DC DC						

Table 4: TEST SEQUENCE FOR THE LIGHTWEIGHT PROFILE (CONTINUED)

es/is_msgo	is/is_msgo	clnp_msgo	clnp_msgo	llc_msgo	inbuf	S _E						
1					x.x.SA.x.x	1						
2						0						
3						4						
4						3						
5						8						
6						9						
7		Ø				10						
8						0						
9						4						
10						5						
11						6						
12	Ø					9						
13						10						
14						0						
15						4						
16						5						
17						7						
18	Ø					9						
19						10						
20						0						
21						4						
22						5						
23						8						
24		Ø				9						
25						10						
26						0						
27						2						
28						3						
29						8						
30		Ø				9						
31						10						
32						0						

VII. CONCLUSIONS AND RECOMMENDATIONS

A. CONTRIBUTIONS OF THIS RESEARCH

The goal of this thesis was to present a series of test sequences for the SAFENET communication protocol. The procedure takes as input high level SAFENET profiles that are specified as a system of communicating machines, and gives as output, complete test sequences for SAFENET's OSI and Lightweight profiles. A brief specification of SAFENET's OSI and Lightweight profiles was given using the system of communicating machines model, and test sequences were generated.

The test method described and employed here further demonstrates the flexibility of the system of communicating machines model. A protocol can be specified, verified, and tested using techniques based on this model. The concept was expanded and applied to a high level profile which encompassed several protocols. In the test procedure, all transition instances in the finite state machine specification is tested in conjunction with each enabling predicate clause. The preliminary steps were employed to determine the input and output variables; the sequence generating procedure was employed to assist in fault coverage. The example test sequences for the OSI and Lightweight profiles were used to demonstrate the application of the specification and testing methods associated with the system of communicating machines model. Since these profiles have the potential for wide spread use in present and future naval combatants, their existence as system of communicating machines model further illustrate the applicability and usefulness of this method. Utilizing a protocol specification method which places emphasis on testing yields better results than using a specification method that was not designed with conformance testing in mind.

Some of the current literature discusses the correctness of a test sequence; their apparent emphasis is on shortening the sequence length. However, the system of communicating machine test procedure emphasizes the ability of the sequence to detect errors rather than the achievement of an optimal test sequence length. This test method can

only test for the presence of desirable behavior in a protocol or profile machine. Given the current level of technology, it is not practical to exhaustively test for the presence of undesirable behavior since all possible errors that could occur in an implementation can not be foreseen.

B. AREAS FOR FURTHER RESEARCH

The issue of security services for data on platforms with a SAFENET implementation exercising data transfers of multiple classifications will have to be addressed. Commercial LANs have encountered and solved this problem with respect to sharing a LAN with a competitor, but with the performance constraints placed upon SAFENET, the completed risk analysis should provide some definitive system configuration with respect to security services. Consequently, research effort must be expended to directly address this issue.

With this test method being as straight forward and easy to apply as it is, it should lend itself very well to automation; research into the feasibility of this could possibly prove very valuable in the wide spread acceptance of this test method. Further research could concentrate on decomposing the protocols within a SAFENET profile and applying the test method. In addition, future research could concentrate on extending the error detection capabilities to detect multiple errors or to detect them in the presence of converging transitions.

APPENDIX

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts _i msg _i	private _i msg _i	xmit_priv _i ate_logica l_msg _i	xmit_clear _i logical_ msg _i	xmit_clear _i map_ msg _i	xmit_clear _i private_ map_ msg _i	xmit_priv _i ate_ msg _i
1 receive	0	x.i.x.x	DC	DC DC							
2 ready	1	DC	DC	DC DC							
3 xmit_time	0	DC	DC	TDA...							
4 sts_msg	2	DC	DC	DC DC	TDA...						
5 cl_trans_msg	3	DC	DC	DC DC							
6 clnp_msg	6	DC	DC	DC DC							
7 llc_msg	17	DC	DC	DC DC							
8 msg_sent	18	DC	DC	DC DC							
9 xmit_clear_msg	0	DC	DC	DC DC							
10 fram_msg	9	DC	DC	DC DC							
11 presentation_msg	14	DC	DC	DC DC							
12 session_msg	15	DC	DC	DC DC							
13 co_trans_msg	16	DC	DC	DC DC							
14 es/is_msg	4	DC	DC	DC DC							
15 llc_msg	17	DC	DC	DC DC							
16 msg_sent	18	DC	DC	DC DC							
17 xmit_clear_msg	0	DC	DC	DC DC							
18 fram_msg	9	DC	DC	DC DC							
19 presentation_msg	14	DC	DC	DC DC							
20 session_msg	15	DC	DC	DC DC							
21 co_trans_msg	16	DC	DC	DC DC							
22 is/is_msg	5	DC	DC	DC DC							
23 llc_msg	17	DC	DC	DC DC							
24 msg_sent	18	DC	DC	DC DC							
25 xmit_clear_msg	0	DC	DC	DC DC							
26 fram_msg	9	DC	DC	DC DC							
27 presentation_msg	14	DC	DC	DC DC							
28 session_msg	15	DC	DC	DC DC							
29 co_trans_msg	16	DC	DC	DC DC							
30 is/is_msg	6	DC	DC	DC DC							
31 llc_msg	17	DC	DC	DC DC							
32 msg_sent	18	DC	DC	DC DC							

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts_ msg _i	private msg _i	xmit_priv ate logical l_msg _i	xmit_clear logical_ msg _i	xmit_clear msg _i	xmit_map_ clear_ map_msg _i	xmit_ private_ map_msg _i	xmit_ ivate_ msg _i
33 xmit_clear_msg_0	0	DC	DC	DC DC				DA.x.FFF.F.F.x				
34 acse_msg_10	10	DC	DC	DC DC								
35 presentation_msg_14	14	DC	DC	DC DC								
36 session_msg_15	15	DC	DC	DC DC								
37 co_trans_msg_16	16	DC	DC	DC DC								
38 es/is_msg_4	4	DC	DC	DC DC								
39 llc_msg_17	17	DC	DC	DC DC								
40 msg_sent_18	18	DC	DC	DC DC								
41 xmit_clear_msg_0	0	DC	DC	DC DC				DA.x.FFF.F.F.x				
42 acse_msg_10	10	DC	DC	DC DC								
43 presentation_msg_14	14	DC	DC	DC DC								
44 session_msg_15	15	DC	DC	DC DC								
45 co_trans_msg_16	16	DC	DC	DC DC								
46 is/is_msg_5	5	DC	DC	DC DC								
47 llc_msg_17	17	DC	DC	DC DC								
48 msg_sent_18	18	DC	DC	DC DC								
49 xmit_clear_msg_0	0	DC	DC	DC DC				DA.x.FFF.F.F.x				
50 acse_msg_10	10	DC	DC	DC DC								
51 presentation_msg_14	14	DC	DC	DC DC								
52 session_msg_15	15	DC	DC	DC DC								
53 co_trans_msg_16	16	DC	DC	DC DC								
54 clnp_msg_6	6	DC	DC	DC DC								
55 llc_msg_17	17	DC	DC	DC DC								
56 msg_sent_18	18	DC	DC	DC DC								
57 xmit_clear_msg_0	0	DC	DC	DC DC				DA.x.FFF.F.F.x				
58 smase_msg_11	11	DC	DC	DC DC								
58 emise_msg_12	12	DC	DC	DC DC								
60 rose_msg_13	13	DC	DC	DC DC								
61 presentation_msg_14	14	DC	DC	DC DC								
62 session_msg_15	15	DC	DC	DC DC								
63 co_trans_msg_16	16	DC	DC	DC DC								
64 is/is_msg_4	4	DC	DC	DC DC								

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts_ msg _i	private_ msg _i	xmit_priv ate logica l_msg _i	xmit_logical_ msg _i	xmit_clear_ clear_ msg _i	xmit_map_ map_ msg _i	xmit_clear_ clear_ msg _i	xmit_private_ private_ map_ msg _i	xmit_private_ private_ msg _i
65 llc_msg	17	DC	DC	DC DC									
66 msg_sent	18	DC	DC	DC DC									
67 xmit_clear_msg	0	DC	DC	DC DC					DA,FFFFFF,x				
68 smase_msg	11	DC	DC	DC DC									
69 cmise_msg	12	DC	DC	DC DC									
70 rose_msg	13	DC	DC	DC DC									
71 presentation_msg	14	DC	DC	DC DC									
72 session_msg	15	DC	DC	DC DC									
73 co_trans_msg	16	DC	DC	DC DC									
74 is/is_msg	5	DC	DC	DC DC									
75 llc_msg	17	DC	DC	DC DC									
76 msg_sent	18	DC	DC	DC DC									
77 xmit_clear_msg	0	DC	DC	DC DC					DA,FFFFFF,x				
78 smase_msg	11	DC	DC	DC DC									
79 cmise_msg	12	DC	DC	DC DC									
80 rose_msg	13	DC	DC	DC DC									
81 presentation_msg	14	DC	DC	DC DC									
82 session_msg	15	DC	DC	DC DC									
83 co_trans_msg	16	DC	DC	DC DC									
84 clnp_msg	6	DC	DC	DC DC									
85 llc_msg	17	DC	DC	DC DC									
86 msg_sent	18	DC	DC	DC DC									
87 xmit_clear_logical_msg	0	DC	DC	DC DC				DA,FFFFFF,x					
88 xmit_clear_map_msg	8	DC	DC	DC DC						DA,FFFFFF,x			
89 flam_msg	9	DC	DC	DC DC									
90 presentation_msg	14	DC	DC	DC DC									
91 session_msg	15	DC	DC	DC DC									
92 co_trans_msg	16	DC	DC	DC DC									
93 es/is_msg	4	DC	DC	DC DC									
94 llc_msg	17	DC	DC	DC DC									
95 msg_sent	18	DC	DC	DC DC									
96 xmit_clear_logical_msg	0	DC	DC	DC DC				DA,FFFFFF,x					

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts_ msg _i	private msg _i	xmit_priv ate logical l_msg _i	xmit_clear logical_ msg _i	xmit_clear map_ msg _i	xmit_ private_ map_msg _i	xmit_pr ivate_ msg _i
97 xmit_clear_map_msg	8	DC	DC	DC DC					DA.FT.FT.Fx		
98 ftam_msg	9	DC	DC	DC DC							
99 presentation_msg	14	DC	DC	DC DC							
100 session_msg	15	DC	DC	DC DC							
101 co_trans_msg	16	DC	DC	DC DC							
102 is/is_msg	5	DC	DC	DC DC							
103 'ic_msg	17	DC	DC	DC DC							
104 msg_sent	18	DC	DC	DC DC							
105 xmit_clear_logical_msg0	0	DC	DC	DC DC			DA.FT.FT.Fx				
106 xmit_clear_map_msg	8	DC	DC	DC DC					DA.FT.FT.Fx		
107 ftam_msg	9	DC	DC	DC DC							
108 presentation_msg	14	DC	DC	DC DC							
109 session_msg	15	DC	DC	DC DC							
110 co_trans_msg	16	DC	DC	DC DC							
111 clnp_msg	6	DC	DC	DC DC							
112 llc_msg	17	DC	DC	DC DC							
113 msg_sent	18	DC	DC	DC DC							
114 xmit_clear_logical_msg0	0	DC	DC	DC DC			DA.FT.FT.Fx				
115 xmit_clear_map_msg	8	DC	DC	DC DC					DA.FT.FT.Fx		
116 acse_msg	10	DC	DC	DC DC							
117 presentation_msg	14	DC	DC	DC DC							
118 session_msg	15	DC	DC	DC DC							
119 co_trans_msg	16	DC	DC	DC DC							
120 es/is_msg	4	DC	DC	DC DC							
121 llc_msg	17	DC	DC	DC DC							
122 msg_sent	18	DC	DC	DC DC							
123 xmit_clear_logical_msg0	0	DC	DC	DC DC			DA.FT.FT.Fx				
124 xmit_clear_map_msg	8	DC	DC	DC DC					DA.FT.FT.Fx		
125 acse_msg	10	DC	DC	DC DC							
126 presentation_msg	14	DC	DC	DC DC							
127 session_msg	15	DC	DC	DC DC							
128 co_trans_msg	16	DC	DC	DC DC							

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts _i msg _i	private _i msg _i	xmit_priv _i ate_logica l_msg _i	xmit_clear _i logical_ msg _i	xmit_clear _i map_ msg _i	xmit_private _i map_ msg _i	xmit_private _i map_ msg _i
129 is/is_msg	5	DC	DC	DC DC							
130 llc_msg	17	DC	DC	DC DC							
131 msg_sent	18	DC	DC	DC DC							
132xmit_clear_logical_msg0	8	DC	DC	DC DC			DA.x.FT.FT.FT.x				
133xmit_clear_map_msg	8	DC	DC	DC DC					DA.x.FT.FT.FT.x		
134 acse_msg	10	DC	DC	DC DC							
135 presentation_msg	14	DC	DC	DC DC							
136 session_msg	15	DC	DC	DC DC							
137 co_trans_msg	16	DC	DC	DC DC							
138 clnp_msg	6	DC	DC	DC DC							
139 llc_msg	17	DC	DC	DC DC							
140 msg_sent	18	DC	DC	DC DC							
141xmit_clear_logical_msg0	8	DC	DC	DC DC			DA.x.FT.FT.FT.x				
142xmit_clear_map_msg	8	DC	DC	DC DC					DA.x.FT.FT.FT.x		
143 smase_msg	11	DC	DC	DC DC							
144 cmise_msg	12	DC	DC	DC DC							
145 rose_msg	13	DC	DC	DC DC							
146 presentation_msg	14	DC	DC	DC DC							
147 session_msg	15	DC	DC	DC DC							
148 co_trans_msg	16	DC	DC	DC DC							
149 es/is_msg	4	DC	DC	DC DC							
150 llc_msg	17	DC	DC	DC DC							
151 msg_sent	18	DC	DC	DC DC							
152xmit_clear_logical_msg0	8	DC	DC	DC DC			DA.x.FT.FT.FT.x				
153xmit_clear_map_msg	8	DC	DC	DC DC					DA.x.FT.FT.FT.x		
154 smase_msg	11	DC	DC	DC DC							
155 cmise_msg	12	DC	DC	DC DC							
156 rose_msg	13	DC	DC	DC DC							
157 presentation_msg	14	DC	DC	DC DC							
158 session_msg	15	DC	DC	DC DC							
159 co_trans_msg	16	DC	DC	DC DC							
160 is/is_msg	5	DC	DC	DC DC							

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts _i msg _i	private _i msg _i	xmit_priv _i logical _i msg _i	xmit_clear _i logical _i msg _i	xmit_clear _i msg _i	xmit_map _i clear _i msg _i	xmit_private _i map _i msg _i	xmit_private _i msg _i
161 llc_msg	17	DC	DC	DC DC								
162 msg_sent	18	DC	DC	DC DC								
163xmit_clear_logical_msg0	0	DC	DC	DC DC				DAx.FTFFTx				
164xmit_clear_map_msg	8	DC	DC	DC DC						DAx.FTFFTx		
165 smase_msg	11	DC	DC	DC DC								
166 cmise_msg	12	DC	DC	DC DC								
167 rose_msg	13	DC	DC	DC DC								
168 presentation_msg	14	DC	DC	DC DC								
169 session_msg	15	DC	DC	DC DC								
170 co_trans_msg	16	DC	DC	DC DC								
171 clnp_msg	6	DC	DC	DC DC								
172 llc_msg	17	DC	DC	DC DC								
173 msg_sent	18	DC	DC	DC DC								
174 private_msg	0	DC	DC	DC DC		DAx.TTTFx						
xmit_private_logical_msg	7	DC	DC	DC DC			DAx.TTTFx					
176xmit_private_map_msg8	8	DC	DC	DC DC							DAx.TTTFx	
177 fram_msg	9	DC	DC	DC DC								
178 presentation_msg	14	DC	DC	DC DC								
179 session_msg	15	DC	DC	DC DC								
180 co_trans_msg	16	DC	DC	DC DC								
181 es/is_msg	4	DC	DC	DC DC								
182 llc_msg	17	DC	DC	DC DC								
183 msg_sent	18	DC	DC	DC DC								
184 private_msg	0	DC	DC	DC DC		DAx.TTTFx						
xmit_private_logical_msg	7	DC	DC	DC DC			DAx.TTTFx					
186xmit_private_map_msg8	8	DC	DC	DC DC							DAx.TTTFx	
187 fram_msg	9	DC	DC	DC DC								
188 presentation_msg	14	DC	DC	DC DC								
189 session_msg	15	DC	DC	DC DC								
190 co_trans_msg	16	DC	DC	DC DC								
191 is/is_msg	5	DC	DC	DC DC								
192 llc_msg	17	DC	DC	DC DC								

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts_ msg _i	private msg _i	xmit_priv ate logical _l_msg _i	xmit_clear logical_ msg _i	xmit_clear map_ msg _i	xmit_ private map_ msg _i	xmit_ ivate_ msg _i
193 msg_sent	18	DC	DC	DC DC							
194 private_msg	0	DC	DC	DC DC		DA.x.T.T.T.F.F.x					
xmit_private_logical_msg	7	DC	DC	DC DC			DA.x.T.T.T.F.F.x				
196mit_private_map_msg	8	DC	DC	DC DC					DA.x.T.T.T.F.F.x		
197 flam_msg	9	DC	DC	DC DC							
198 presentation_msg	14	DC	DC	DC DC							
199 session_msg	15	DC	DC	DC DC							
200 co_trans_msg	16	DC	DC	DC DC							
201 clnp_msg	6	DC	DC	DC DC							
202 llc_msg	17	DC	DC	DC DC							
203 msg_sent	18	DC	DC	DC DC							
204 private_msg	0	DC	DC	DC DC		DA.x.T.T.T.F.F.x					
xmit_private_logical_msg	7	DC	DC	DC DC			DA.x.T.T.T.F.F.x				
206mit_private_map_msg	8	DC	DC	DC DC					DA.x.T.T.T.F.F.x		
207 acse_msg	10	DC	DC	DC DC							
208 presentation_msg	14	DC	DC	DC DC							
209 session_msg	15	DC	DC	DC DC							
210 co_trans_msg	16	DC	DC	DC DC							
211 es/is_msg	4	DC	DC	DC DC							
212 llc_msg	17	DC	DC	DC DC							
213 msg_sent	18	DC	DC	DC DC							
214 private_msg	0	DC	DC	DC DC		DA.x.T.T.T.F.F.x					
xmit_private_logical_msg	7	DC	DC	DC DC			DA.x.T.T.T.F.F.x				
216mit_private_map_msg	8	DC	DC	DC DC					DA.x.T.T.T.F.F.x		
217 acse_msg	10	DC	DC	DC DC							
218 presentation_msg	14	DC	DC	DC DC							
219 session_msg	15	DC	DC	DC DC							
220 co_trans_msg	16	DC	DC	DC DC							
221 is/is_msg	5	DC	DC	DC DC							
222 llc_msg	17	DC	DC	DC DC							
223 msg_sent	18	DC	DC	DC DC							
224 private_msg	0	DC	DC	DC DC		DA.x.T.T.T.F.F.x					

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sis_ msg _i	private_ msg _i	xmit_priv ate logical l_msg _i	xmit_clear logical_ msg _i	xmit_clear_ map_ msg _i	xmit_priv ate logical_ map_ msg _i	xmit_priv ate_ map_ msg _i
xmit_private_logical_msg	7	DC	DC	DC			DA,x.TTFTT,x				
xmit_private_map_msg	8	DC	DC	DC						DA,x.TTFTT,x	
acse_msg	10	DC	DC	DC							
presentation_msg	14	DC	DC	DC							
session_msg	15	DC	DC	DC							
co_trans_msg	16	DC	DC	DC							
clnp_msg	6	DC	DC	DC							
llc_msg	17	DC	DC	DC							
msg_sent	18	DC	DC	DC							
private_msg	0	DC	DC	DC		DA,x.TTFTT,x					
xmit_private_logical_msg	7	DC	DC	DC			DA,x.TTFTT,x				
xmit_private_map_msg	8	DC	DC	DC						DA,x.TTFTT,x	
smase_msg	11	DC	DC	DC							
cmise_msg	12	DC	DC	DC							
rose_msg	13	DC	DC	DC							
presentation_msg	14	DC	DC	DC							
session_msg	15	DC	DC	DC							
co_trans_msg	16	DC	DC	DC							
es/is_msg	4	DC	DC	DC							
llc_msg	17	DC	DC	DC							
msg_sent	18	DC	DC	DC							
private_msg	0	DC	DC	DC		DA,x.TTFTT,x					
xmit_private_logical_msg	7	DC	DC	DC			DA,x.TTFTT,x				
xmit_private_map_msg	8	DC	DC	DC						DA,x.TTFTT,x	
smase_msg	11	DC	DC	DC							
cmise_msg	12	DC	DC	DC							
rose_msg	13	DC	DC	DC							
presentation_msg	14	DC	DC	DC							
session_msg	15	DC	DC	DC							
co_trans_msg	16	DC	DC	DC							
is/is_msg	5	DC	DC	DC							
llc_msg	17	DC	DC	DC							

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts_ msg _i	private msg _i	xmit_priv ate logical _l_msg _i	xmit_clear logical_ msg _i	xmit_clear_ map_ msg _i	xmit_private map msg _i	xmit_private msg _i
257 msg_sent	18	DC	DC	DC DC							
258 private_msg	0	DC	DC	DC DC		DA.x.T.T.F.F.T.x					
xmit_private_logical_msg	7	DC	DC	DC DC			DA.x.T.T.F.F.T.x				
260xmit_private_map_msg	8	DC	DC	DC DC						DA.x.T.T.F.F.T.x	
261 smase_msg	11	DC	DC	DC DC							
262 cmise_msg	12	DC	DC	DC DC							
263 rose_msg	13	DC	DC	DC DC							
264 presentation_msg	14	DC	DC	DC DC							
265 session_msg	15	DC	DC	DC DC							
266 co_trans_msg	16	DC	DC	DC DC							
267 clnp_msg	6	DC	DC	DC DC							
268 llc_msg	17	DC	DC	DC DC							
269 msg_sent	18	DC	DC	DC DC							
270 private_msg	0	DC	DC	DC DC		DA.x.T.T.F.F.F.F.x					
271xmit_private_msg	7	DC	DC	DC DC						DA.x.T.T.F.F.F.F.x	
272 ftam_msg	9	DC	DC	DC DC							
273 presentation_msg	14	DC	DC	DC DC							
274 session_msg	15	DC	DC	DC DC							
275 co_trans_msg	16	DC	DC	DC DC							
276 es/is_msg	4	DC	DC	DC DC							
277 llc_msg	17	DC	DC	DC DC							
278 msg_sent	18	DC	DC	DC DC							
279 private_msg	0	DC	DC	DC DC		DA.x.T.T.F.F.F.F.x					
280xmit_private_msg	7	DC	DC	DC DC						DA.x.T.T.F.F.F.F.x	
281 ftam_msg	9	DC	DC	DC DC							
282 presentation_msg	14	DC	DC	DC DC							
283 session_msg	15	DC	DC	DC DC							
284 co_trans_msg	16	DC	DC	DC DC							
285 is/is_msg	5	DC	DC	DC DC							
286 llc_msg	17	DC	DC	DC DC							
287 msg_sent	18	DC	DC	DC DC							
288 private_msg	0	DC	DC	DC DC		DA.x.T.T.F.F.F.F.x					

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts_ msg _i	private_ msg _i	xmit_priv ate_logica l_msg _i	xmit_clear_ logical_ msg _i	xmit_clear_ map_ msg _i	xmit_ private_ map_ msg _i	xmit_ private_ map_ msg _i
289 xmit_private_msg	7	DC	DC	DC	DC						DA,TF,TFE
290 flam_msg	9	DC	DC	DC	DC						
291 presentation_msg	14	DC	DC	DC	DC						
292 session_msg	15	DC	DC	DC	DC						
293 co_trans_msg	16	DC	DC	DC	DC						
294 clnp_msg	6	DC	DC	DC	DC						
295 llc_msg	17	DC	DC	DC	DC						
295 msg_sent	18	DC	DC	DC	DC						
297 private_msg	0	DC	DC	DC	DC	DA,TF,TFE,x					
298 xmit_private_msg	7	DC	DC	DC	DC					DA,TF,TFE	
299 acse_msg	10	DC	DC	DC	DC						
300 presentation_msg	14	DC	DC	DC	DC						
301 session_msg	15	DC	DC	DC	DC						
302 co_trans_msg	16	DC	DC	DC	DC						
303 es/is_msg	4	DC	DC	DC	DC						
304 llc_msg	17	DC	DC	DC	DC						
305 msg_sent	18	DC	DC	DC	DC						
306 private_msg	0	DC	DC	DC	DC	DA,TF,TFE,x					
307 xmit_private_msg	7	DC	DC	DC	DC					DA,TF,TFE	
308 acse_msg	10	DC	DC	DC	DC						
309 presentation_msg	14	DC	DC	DC	DC						
310 session_msg	15	DC	DC	DC	DC						
311 co_trans_msg	16	DC	DC	DC	DC						
312 is/is_msg	5	DC	DC	DC	DC						
313 llc_msg	17	DC	DC	DC	DC						
314 msg_sent	18	DC	DC	DC	DC						
315 private_msg	0	DC	DC	DC	DC	DA,TF,TFE,x					
316 xmit_private_msg	7	DC	DC	DC	DC					DA,TF,TFE	
317 acse_msg	10	DC	DC	DC	DC						
318 presentation_msg	14	DC	DC	DC	DC						
319 session_msg	15	DC	DC	DC	DC						
320 co_trans_msg	16	DC	DC	DC	DC						

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S1	MEDIUM1	Transfer1	outbuf1 (1,2)	sts_ msg1	private_ msg1	xmit_priv ate logical l_msg1	xmit_clear logical_ msg1	xmit_clear map_msg1	xmit_ private_ map_msg1	xmit_priv ate logical l_msg1	xmit_clear map_msg1	xmit_ private_ map_msg1	xmit_priv ate logical l_msg1	xmit_clear map_msg1	xmit_ private_ map_msg1	xmit_priv ate logical l_msg1	xmit_clear map_msg1	xmit_ private_ map_msg1
321 clnp_msg	6	DC		DC DC															
322 llc_msg	17	DC		DC DC															
323 msg_sent	18	DC		DC DC															
324 private_msg	0	DC		DC DC		DA.x.T.F.F.F.T.*													
325 xmit_private_msg	7	DC		DC DC													DA.x.T.F.F.F.T.*		
326 smase_msg	11	DC		DC DC															
327 cmise_msg	12	DC		DC DC															
328 rose_msg	13	DC		DC DC															
329 presentation_msg	14	DC		DC DC															
330 session_msg	15	DC		DC DC															
331 co_trans_msg	16	DC		DC DC															
332 es/is_msg	4	DC		DC DC															
333 llc_msg	17	DC		DC DC															
334 msg_sent	18	DC		DC DC															
335 private_msg	0	DC		DC DC		DA.x.T.F.F.F.T.*													
336 xmit_private_msg	7	DC		DC DC													DA.x.T.F.F.F.T.*		
337 smase_msg	11	DC		DC DC															
338 cmise_msg	12	DC		DC DC															
339 rose_msg	13	DC		DC DC															
340 presentation_msg	14	DC		DC DC															
341 session_msg	15	DC		DC DC															
342 co_trans_msg	16	DC		DC DC															
343 es/is_msg	5	DC		DC DC															
344 llc_msg	17	DC		DC DC															
345 msg_sent	18	DC		DC DC															
346 private_msg	0	DC		DC DC		DA.x.T.F.F.F.T.*													
347 xmit private msg	7	DC		DC DC													DA.x.T.F.F.F.T.*		
348 smase_msg	11	DC		DC DC															
349 cmise_msg	12	DC		DC DC															
350 rose_msg	13	DC		DC DC															
351 presentation msg	14	DC		DC DC															
352 session_msg	15	DC		DC DC															

Table 2: TEST SEQUENCE FOR THE OSI PROFILE

Transition	S _i	MEDIUM _i	Transfer _i	outbuf _i (1,2)	sts_ msg _i	private_ msg _i	xmit_priv ate_logica l_msg _i	xmit_clear logical_ msg _i	xmit_clear msg _i	xmit_map_ msg _i	xmit_clear msg _i	xmit_ private_ map_msg _i	xmit_ ivate_ msg _i
353 co_trans_msg		DC	DC	DC DC									
354 clnp_msg		DC	DC	DC DC									
355 llc_msg		DC	DC	DC DC									
356 msg_sent		DC	DC	DC DC									
357													
358													
359													
360													
361													
362													
363													
364													
365													
366													
367													
368													
369													
370													
371													
372													
373													
374													
375													
376													
377													
378													
379													
380													
381													
382													
383													
384													

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msgl	acse msgl	smase_ msgl	cmise_ msgl	rose_ msgl	presen- tation_ msgl	session_ msgl	co trans_ msgl	cl trans_ msgl	es/is_ msgl	is/is_ msgl	clnp_ msgl	llc_ msgl
1												
2												
3												
4												
5								TDA.x.EFT				
6											TDA.x.x.x	
7												TDA.x.x.x
8												
9												
10 xDA.x.x.x												
11					xDA.x.x.x							
12						xDA.x.x.x						
13							xDA.x.TEFx					
14									xDA.x.x.x			xDA.x.x.x
15												
16												
17												
18 xDA.x.x.x												
19					xDA.x.x.x							
20						xDA.x.x.x						
21							xDA.x.TEFx					
22										xDA.x.x.x		
23												xDA.x.x.x
24												
25												
26 xDA.x.x.x												
27					xDA.x.x.x							
28						xDA.x.x.x						
29							xDA.x.EFT.x					
30											xDA.x.x.x	
31												xDA.x.x.x
32												

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msg _i	acse_ msg _i	smase_ msg _i	cmise_ msg _i	rose_ msg _i	presen- tation_ msg _i	session_ msg _i	co trans_ msg _i	cl trans_ msg _i	es/is_ msg _i	is/is_ msg _i	clnp_ msg _i	llc_ msg _i
33												
34	x.DA.x.x.x											
35				x.DA.x.x.x								
36						x.DA.x.x.x						
37							x.DA.x.TFFx		x.DA.x.x.x			x.DA.x.x.x
38												
39												
40												
41												
42	x.DA.x.x.x											
43				x.DA.x.x.x								
44						x.DA.x.x.x						
45							x.DA.x.FTfx			x.DA.x.x.x		
46												
47												x.DA.x.x.x
48												
49												
50	x.DA.x.x.x											
51				x.DA.x.x.x								
52						x.DA.x.x.x						
53							x.DA.x.FTfx					
54											x.DA.x.x.x	x.DA.x.x.x
55												
56												
57												
58		x.DA.x.x.x										
58		x.DA.x.x.x										
60			x.DA.x.x.x									
61				x.DA.x.x.x								
62						x.DA.x.x.x						
63							x.DA.x.TFFx					
64									x.DA.x.x.x			

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msgl	acse_ msgl	smase_ msgl	cmise_ msgl	rose_ msgl	presen- tation_ msgl	session_ msgl	co_trans_ msgl	cl_trans_ msgl	es/is_ msgl	is/is_ msgl	clnp_ msgl	llc_ msgl
65												
66												
67												
68		xDA.x.x.x										
69			xDA.x.x.x									
70				xDA.x.x.x								
71					xDA.x.x.x							
72						xDA.x.x.x						
73							xDA.x.F.T.F.x					
74										xDA.x.x.x		xDA.x.x.x
75												
76												
77												
78		xDA.x.x.x										
79			xDA.x.x.x									
80				xDA.x.x.x								
81					xDA.x.x.x							
82						xDA.x.x.x						
83							xDA.x.F.F.T.x				xDA.x.x.x	
84												xDA.x.x.x
85												
86												
87												
88												
89 xDA.x.x.x												
90					xDA.x.x.x							
91						xDA.x.x.x						
92							xDA.x.T.F.F.x					
93										xDA.x.x.x		
94												xDA.x.x.x
95												
96												

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msg _i	acce_ msg _i	smase_ msg _i	cmise_ msg _i	rose_ msg _i	presen- tation_ msg _i	session_ msg _i	co trans_ msg _i	cl_trans_ msg _i	es/is_ msg _i	is/is_ msg _i	clnp_ msg _i	llc_ msg _i
97												
98 1.DA.1.1.1												
99					1.DA.1.1.1							
100						1.DA.1.1.1						
101							1.DA.1.FTF.1					
102										1.DA.1.1.1		
103												1.DA.1.1.1
104												
105												
106												
107 1.DA.1.1.1												
108					1.DA.1.1.1							
109						1.DA.1.1.1						
110							1.DA.1.EFT.1				1.DA.1.1.1	
111												
112												
113												
114												
115												
116 1.DA.1.1.1												
117					1.DA.1.1.1							
118						1.DA.1.1.1						
119							1.DA.1.TFF.1					
120									1.DA.1.1.1			
121												1.DA.1.1.1
122												
123												
124												
125 1.DA.1.1.1												
126					1.DA.1.1.1							
127						1.DA.1.1.1						
128							1.DA.1.FTF.1					

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msgi	acse msgi	smase_ msgi	cmise_ msgi	rose_ msgi	presen- tation_ msgi	session_ msgi	co trans_ msgi	cl trans_ msgi	es/is_ msgi	is/is_ msgi	clnp_ msgi	llc_ msgi
129										x.DA.x.x.x		x.DA.x.x.x
130												
131												
132												
133												
134	x.DA.x.x.x											
135					x.DA.x.x.x							
136						x.DA.x.x.x						
137							x.DA.x.F.F.F.x					
138											x.DA.x.x.x	
139												x.DA.x.x.x
140												
141												
142												
143		x.DA.x.x.x										
144			x.DA.x.x.x									
145				x.DA.x.x.x								
146					x.DA.x.x.x							
147						x.DA.x.x.x						
148							X.DA.x.T.F.F.x					
149												
150									x.DA.x.x.x			x.DA.x.x.x
151												
152												
153												
154		x.DA.x.x.x										
155			x.DA.x.x.x									
156				x.DA.x.x.x								
157					x.DA.x.x.x							
158												
159							x.DA.x.F.T.F.x					
160										x.DA.x.x.x		

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

flam_ msgi	acse msgi	smase_ msgi	cmise_ msgi	prese- ntation_ msgi	session_ msgi	co trans_ msgi	cl_trans_ msgi	es/is_ msgi	is/is_ msgi	clnp_ msgi	llc_ msgi
161											1.DA.1.1.1
162											
163											
164											
165		1.DA.1.1.1									
166			1.DA.1.1.1								
167				1.DA.1.1.1							
168					1.DA.1.1.1						
169						1.DA.1.1.1					
170						1.DA.1.1.1					
171										1.DA.1.1.1	
172											
173											
174											
175											
176											
177	1.DA.1.1.1										
178				1.DA.1.1.1							
179					1.DA.1.1.1						
180						1.DA.1.1.1					
181											
182								1.DA.1.1.1			
183											1.DA.1.1.1
184											
185											
186											
187	1.DA.1.1.1										
188				1.DA.1.1.1							
189					1.DA.1.1.1						
190						1.DA.1.1.1					
191									1.DA.1.1.1		
192											1.DA.1.1.1

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msg1	acse msg1	smase msg1	cmise msg1	rose msg1	presen- tation_ msg1	session_ msg1	co trans_ msg1	cl trans_ msg1	es/is_ msg1	is/is_ msg1	clnp_ msg1	llc_ msg1
193												
194												
195												
196												
197	xDA.x.x.x											
198					xDA.x.x.x							
199						xDA.x.x.x						
200							xDA.x.FFT.x				xDA.x.x.x	
201												
202												
203												
204												
205												
206												
207	xDA.x.x.x											
208					xDA.x.x.x							
209						xDA.x.x.x						
210							xDA.x.TFE.x					
211									xDA.x.x.x			
212												xDA.x.x.x
213												
214												
215												
216												
217	xDA.x.x.x											
218					xDA.x.x.x							
219						xDA.x.x.x						
220							xDA.x.FTF.x					
221										xDA.x.x.x		
222												xDA.x.x.x
223												
224												

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msgi	acse msgi	smase_ msgi	cmise_ msgi	rose_ msgi	presen- tation_ msgi	session_ msgi	co trans_ msgi	cl_trans_ msgi	es/is_ msgi	is/is_ msgi	clnp_ msgi	llc_ msgi
225												
226												
227	xDA,x,x,x											
228					xDA,x,x,x							
229						xDA,x,x,x						
230							xDA,X,FFTx					
231											xDA,x,x,x	xDA,x,x,x
232												
233												
234												
235												
236												
237		xDA,x,x,x										
238			xDA,x,x,x									
239				xDA,x,x,x								
240					xDA,x,x,x							
241						xDA,x,x,x						
242							xDA,x,TFEx					
243									xDA,x,x,x			xDA,x,x,x
244												
245												
246												
247												
248												
249		xDA,x,x,x										
250			xDA,x,x,x									
251				xDA,x,x,x								
252					xDA,x,x,x							
253						xDA,x,x,x						
254							xDA,x,TFEx					
255										xDA,x,x,x		
256												xDA,x,x,x

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msgl	acse msgl	smase_ msgl	cmise_ msgl	rose_ msgl	presen- tation_ msgl	session_ msgl	co trans_ msgl	cl trans_ msgl	es/is_ msgl	is/is_ msgl	clnp_ msgl	llc_ msgl
257												
258												
259												
260												
261		xDA,x,x,x										
262			xDA,x,x,x									
263				xDA,x,x,x								
264					xDA,x,x,x							
265						xDA,x,x,x						
266							xDA,x,FEFx					
267											xDA,x,x,x	xDA,x,x,x
268												
269												
270												
271												
272xDA,x,x,x												
273					xDA,x,x,x							
274												
275						xDA,x,x,x	xDA,x,TEFx					
276									xDA,x,x,x			xDA,x,x,x
277												
278												
279												
280												
281xDA,x,x,x												
282					xDA,x,x,x							
283						xDA,x,x,x						
284							xDA,x,FEFx					
285										xDA,x,x,x		xDA,x,x,x
286												
287												
288												

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

fram_ msg1	acse msg1	smase msg1	cmise msg1	prese- ntation_ msg1	session_ msg1	co trans_ msg1	cl_trans_ msg1	es/is_ msg1	is/is_ msg1	clnp_ msg1	llc_ msg1
289											
290	xDA,x,x,x			xDA,x,x,x							
291											
292					xDA,x,x,x						
293						xDA,x,FFTx					
294										xDA,x,x,x	
295											xDA,x,x,x
296											
297											
298											
299	xDA,x,x,x										
300				xDA,x,x,x							
301					xDA,x,x,x						
302						xDA,x,TFFx					
303								xDA,x,x,x			
304											xDA,x,x,x
305											
306											
307											
308	xDA,x,x,x										
309				xDA,x,x,x							
310					xDA,x,x,x						
311						xDA,x,FTFx					
312									xDA,x,x,x		
313											xDA,x,x,x
314											
315											
316											
317	xDA,x,x,x										
318				xDA,x,x,x							
319					xDA,x,x,x						
320						xDA,x,FFTx					

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msg1	acse_ msg1	smase_ msg1	cmise_ msg1	rose_ msg1	presen- tation_ msg1	session_ msg1	co trans_ msg1	cl trans_ msg1	es/is_ msg1	is/is_ msg1	clnp_ msg1	llc_ msg1
321											xDA,x,x,x	xDA,x,x,x
322												
323												
324												
325												
326		xDA,x,x,x										
327			xDA,x,x,x									
328				xDA,x,x,x								
329					xDA,x,x,x							
330						xDA,x,x,x						
331							xDA,x,TEF,x					
332									xDA,x,x,x			
333												xDA,x,x,x
334												
335												
336												
337		xDA,x,x,x										
338			xDA,x,x,x									
339				xDA,x,x,x								
340					xDA,x,x,x							
341						xDA,x,x,x						
342							xDA,x,TEF,x					
343										xDA,x,x,x		
344												xDA,x,x,x
345												
346												
347												
348		xDA,x,x,x										
349			xDA,x,x,x									
350				xDA,x,x,x								
351					xDA,x,x,x							
352						xDA,x,x,x						

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

ftam_ msg1	acse msg1	smase_ msg1	cmise_ msg1	rose_ msg1	presen tation_ msg1	session_ msg1	co_trans_ msg1	cl_trans_ msg1	es/is msg1	is/is msg1	clnp_ msg1	llc_ msg1
353							x.DA.x.FF.T.x				x.DA.x.x.x	x.DA.x.x.x
354												
355												
356												
357												
358												
359												
360												
361												
362												
363												
364												
365												
366												
367												
368												
369												
370												
371												
372												
373												
374												
375												
376												
377												
378												
379												
380												
381												
382												
382												
384												

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbufo (1,2)	sts_ msgo	private_ msgo	xmit_priv ate_logica l_msgo	xmit_clear logical_ msgo	xmit_clear_ msgo	xmit_private_ map_msgo	xmit_priv ate_ msgo	ftam_ msgo
1	DC	DC	DC							
2	Ø	DC	DC							
3	DC	T ₁ SA _{1,1}	Ø							
4	DC	T ₁ SA _{1,1}	DC	Ø						
5	DC	T ₁ SAFF _{1,1}	DC	DC						
6	DC	T ₁ SA _{1,1}	DC	DC						
7	DC	T ₁ SA _{1,1}	DC	DC						
8	Ø	DC	DC	DC						
9	DC	T ₁ SAFF _{1,1}	DC	DC			Ø			
10	DC	T ₁ SA _{1,1}	DC	DC						Ø
11	DC	T ₁ SA _{1,1}	DC	DC						
12	DC	T ₁ SA _{1,1}	DC	DC						
13	DC	T ₁ SAFF _{1,1}	DC	DC						
14	DC	T ₁ SA _{1,1}	DC	DC						
15	DC	T ₁ SA _{1,1}	DC	DC						
16	Ø	DC	DC	DC						
17	DC	T ₁ SAFF _{1,1}	DC	DC			Ø			Ø
18	DC	T ₁ SA _{1,1}	DC	DC						
19	DC	T ₁ SA _{1,1}	DC	DC						
20	DC	T ₁ SA _{1,1}	DC	DC						
21	DC	T ₁ SAFF _{1,1}	DC	DC						
22	DC	T ₁ SA _{1,1}	DC	DC						
23	DC	T ₁ SA _{1,1}	DC	DC						
24	Ø	DC	DC	DC						
25	DC	T ₁ SAFF _{1,1}	DC	DC			Ø			Ø
26	DC	T ₁ SA _{1,1}	DC	DC						
27	DC	T ₁ SA _{1,1}	DC	DC						
28	DC	T ₁ SA _{1,1}	DC	DC						
29	DC	T ₁ SAFF _{1,1}	DC	DC						
30	DC	T ₁ SA _{1,1}	DC	DC						
31	DC	T ₁ SA _{1,1}	DC	DC						
32	Ø	DC	DC	DC						

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbufo (1,2)	sts_ msgo	private_ msgo	xmit_priv ate_logical_ l_msgo	xmit_clear logical_ msgo	xmit_clear map_ msgo	xmit_ private_ map_ msgo	xmit_ ivate_ msgo	fram_ msgo
33 DC	1.SAFFETEx	DC DC								
34 DC	1.SA.SA.SA	DC DC				Ø				
35 DC	1.SA.SA.SA	DC DC								
36 DC	1.SA.SA.SA	DC DC								
37 DC	1.SA.SA.TEEx	DC DC								
38 DC	1.SA.SA.SA	DC DC								
39 DC	1.SA.SA.SA	DC DC								
40 Ø	DC	DC DC				Ø				
41 DC	1.SAFFETEx	DC DC								
42 DC	1.SA.SA.SA	DC DC								
43 DC	1.SA.SA.SA	DC DC								
44 DC	1.SA.SA.SA	DC DC								
45 DC	1.SA.SA.TEEx	DC DC								
46 DC	1.SA.SA.SA	DC DC								
47 DC	1.SA.SA.SA	DC DC								
48 Ø	DC	DC DC								
49 DC	1.SAFFETEx	DC DC				Ø				
50 DC	1.SA.SA.SA	DC DC								
51 DC	1.SA.SA.SA	DC DC								
52 DC	1.SA.SA.SA	DC DC								
53 DC	1.SA.SA.TEEx	DC DC								
54 DC	1.SA.SA.SA	DC DC								
55 DC	1.SA.SA.SA	DC DC								
56 Ø	DC	DC DC								
57 DC	1.SAFFETEx	DC DC				Ø				
58 DC	1.SA.SA.SA	DC DC								
59 DC	1.SA.SA.SA	DC DC								
60 DC	1.SA.SA.SA	DC DC								
61 DC	1.SA.SA.SA	DC DC								
62 DC	1.SA.SA.SA	DC DC								
63 DC	1.SA.SA.TEEx	DC DC								
64 DC	1.SA.SA.SA	DC DC								

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outhbu ₀ (1,2)	sts_ msg ₀	private_ msg ₀	xmit_priv ate_logica l_msg ₀	xmit_clear logical_ msg ₀	xmit_clear_ map_ msg ₀	xmit_ private_ map_ msg ₀	xmit_ivate_ msg ₀	ftam_ msg ₀
65 DC	xxSA,xx	DC DC								
66 Ø	DC	DC DC								
67 DC	xxSA,FEFF,xx	DC DC				Ø				
68 DC	xxSA,xx	DC DC								
69 DC	xxSA,xx	DC DC								
70 DC	xxSA,xx	DC DC								
71 DC	xxSA,xx	DC DC								
72 DC	xxSA,xx	DC DC								
73 DC	xxSA,FEFF,xx	DC DC								
74 DC	xxSA,xx	DC DC								
75 DC	xxSA,xx	DC DC								
76 Ø	DC	DC DC								
77 DC	xxSA,FEFF,xx	DC DC				Ø				
78 DC	xxSA,xx	DC DC								
79 DC	xxSA,xx	DC DC								
80 DC	xxSA,xx	DC DC								
81 DC	xxSA,xx	DC DC								
82 DC	xxSA,xx	DC DC								
83 DC	xxSA,FEFF,xx	DC DC								
84 DC	xxSA,xx	DC DC								
85 DC	xxSA,xx	DC DC								
86 Ø	DC	DC DC								
87 DC	xxSA,FEFF,xx	DC DC				Ø				
88 DC	xxSA,FEFF,xx	DC DC					Ø			
89 DC	xxSA,xx	DC DC								Ø
90 DC	xxSA,xx	DC DC								
91 DC	xxSA,xx	DC DC								
92 DC	xxSA,FEFF,xx	DC DC								
93 DC	xxSA,xx	DC DC								
94 DC	xxSA,xx	DC DC								
95 Ø	DC	DC DC				Ø				
96 DC	xxSA,FEFF,xx	DC DC								

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfero	outbufo (1,2)	sts_ msgo	private_ msgo	xmit_priv ate_logica l_msgo	xmit_clear logical_ msgo	xmit_clear_ msgo	xmit_private_ map_msgo	xmit_private_ map_msgo	xmit_private_ map_msgo	ftam_ msgo
97	DC	1.SA.F.T.F.F.	DC	DC							
98	DC	1.SA.S.A.S.	DC	DC							
99	DC	1.SA.S.A.S.	DC	DC							
100	DC	1.SA.S.A.S.	DC	DC							
101	DC	1.SA.F.T.F.F.	DC	DC							
102	DC	1.SA.S.A.S.	DC	DC							
103	DC	1.SA.S.A.S.	DC	DC							
104	Ø	DC	DC	DC							
105	DC	1.SA.F.T.F.F.	DC	DC		Ø					
106	DC	1.SA.F.T.F.F.	DC	DC				Ø			
107	DC	1.SA.S.A.S.	DC	DC							Ø
108	DC	1.SA.S.A.S.	DC	DC							
109	DC	1.SA.S.A.S.	DC	DC							
110	DC	1.SA.F.F.T.F.	DC	DC							
111	DC	1.SA.S.A.S.	DC	DC							
112	DC	1.SA.S.A.S.	DC	DC							
113	Ø	DC	DC	DC							
114	DC	1.SA.F.T.F.F.	DC	DC		Ø					
115	DC	1.SA.F.T.F.F.	DC	DC				Ø			
116	DC	1.SA.S.A.S.	DC	DC							
117	DC	1.SA.S.A.S.	DC	DC							
118	DC	1.SA.S.A.S.	DC	DC							
119	DC	1.SA.T.F.F.F.	DC	DC							
120	DC	1.SA.S.A.S.	DC	DC							
121	DC	1.SA.S.A.S.	DC	DC							
122	Ø	DC	DC	DC							
123	DC	1.SA.F.T.F.F.	DC	DC		Ø					
124	DC	1.SA.F.T.F.F.	DC	DC				Ø			
125	DC	1.SA.S.A.S.	DC	DC							
126	DC	1.SA.S.A.S.	DC	DC							
127	DC	1.SA.S.A.S.	DC	DC							
128	DC	1.SA.F.T.F.F.	DC	DC							

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbufo (1,2)	sts_ msgo	private_ msgo	xmit_priv ate_logica l_msgo	xmit_clear logical_ msgo	xmit_clear_ map_msgo	xmit_ private_ map_msgo	xmit_priv ivate_ msgo	ftam_ msgo
129 DC	1.1.SA.1.1	DC DC								
130 DC	1.1.SA.1.1	DC DC								
131 Ø	DC	DC DC								
132 DC	1.SA.FT.FT.1	DC DC			Ø					
133 DC	1.SA.FT.FT.1	DC DC					Ø			
134 DC	1.1.SA.1.1	DC DC								
135 DC	1.1.SA.1.1	DC DC								
136 DC	1.1.SA.1.1	DC DC								
137 DC	1.1.SA.FT.1	DC DC								
138 DC	1.1.SA.1.1	DC DC								
139 DC	1.1.SA.1.1	DC DC								
140 Ø	DC	DC DC			Ø					
141 DC	1.SA.FT.FT.1	DC DC								
142 DC	1.SA.FT.FT.1	DC DC					Ø			
143 DC	1.1.SA.1.1	DC DC								
144 DC	1.1.SA.1.1	DC DC								
145 DC	1.1.SA.1.1	DC DC								
146 DC	1.1.SA.1.1	DC DC								
147 DC	1.1.SA.1.1	DC DC								
148 DC	1.1.SA.FT.1	DC DC								
149 DC	1.1.SA.1.1	DC DC								
150 DC	1.1.SA.1.1	DC DC								
151 Ø	DC	DC DC								
152 DC	1.SA.FT.FT.1	DC DC			Ø					
153 DC	1.SA.FT.FT.1	DC DC					Ø			
154 DC	1.1.SA.1.1	DC DC								
155 DC	1.1.SA.1.1	DC DC								
156 DC	1.1.SA.1.1	DC DC								
157 DC	1.1.SA.1.1	DC DC								
158 DC	1.1.SA.1.1	DC DC								
159 DC	1.1.SA.FT.1	DC DC								
160 DC	1.1.SA.1.1	DC DC								

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbufo (1,2)	sts_ msg ₀	private_ msg ₀	xmit_priv ate logical l_msg ₀	xmit_clear logical_ msg ₀	xmit_clear_ msg ₀	xmit_ private_ map_msg ₀	xmit_ private_ map_msg ₀	ftam_ msg ₀
161	DC	1.1.SA.1.1	DC DC							
162	Ø	DC	DC DC							
163	DC	1.SA.FT.FT.1	DC DC			Ø				
164	DC	1.SA.FT.FT.1	DC DC				Ø			
165	DC	1.1.SA.1.1	DC DC							
166	DC	1.1.SA.1.1	DC DC							
167	DC	1.1.SA.1.1	DC DC							
168	DC	1.1.SA.1.1	DC DC							
169	DC	1.1.SA.1.1	DC DC							
170	DC	1.1.SA.FT.1	DC DC							
171	DC	1.1.SA.1.1	DC DC							
172	DC	1.1.SA.1.1	DC DC							
173	Ø	DC	DC DC							
174	DC	1.SA.TT.TFF.1	DC DC	Ø						
175	DC	1.SA.TT.TFF.1	DC DC		Ø					
176	DC	1.SA.TT.TFF.1	DC DC					Ø		
177	DC	1.1.SA.1.1	DC DC							Ø
178	DC	1.1.SA.1.1	DC DC							
179	DC	1.1.SA.1.1	DC DC							
180	DC	1.1.SA.TFF.1	DC DC							
181	DC	1.1.SA.1.1	DC DC							
182	DC	1.1.SA.1.1	DC DC							
183	Ø	DC	DC DC							
184	DC	1.SA.TT.TFF.1	DC DC	Ø						
185	DC	1.SA.TT.TFF.1	DC DC		Ø					
186	DC	1.SA.TT.TFF.1	DC DC					Ø		
187	DC	1.1.SA.1.1	DC DC							Ø
188	DC	1.1.SA.1.1	DC DC							
189	DC	1.1.SA.1.1	DC DC							
190	DC	1.1.SA.FT.1	DC DC							
191	DC	1.1.SA.1.1	DC DC							
192	DC	1.1.SA.1.1	DC DC							

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbuf ₀ (1,2)	sts_ msg ₀	private_ msg ₀	xmit_priv ate logical l_msg ₀	xmit_clear logical_ msg ₀	xmit_ clear_ msg ₀	xmit_ private_ map_msg ₀	xmit_priv ivate_ msg ₀	ftam_ msg ₀
193	DC	DC DC								
194	DC	1.SA.TTFFx	DC DC	Ø						
195	DC	1.SA.TTFFx	DC DC		Ø					
196	DC	1.SA.TTFFx	DC DC					Ø		
197	DC	1.1.SA.x.x	DC DC							Ø
198	DC	1.1.SA.x.x	DC DC							
199	DC	1.1.SA.x.x	DC DC							
200	DC	1.1.SA.FFTx	DC DC							
201	DC	1.1.SA.x.x	DC DC							
202	DC	1.1.SA.x.x	DC DC							
203	Ø	DC	DC DC							
204	DC	1.SA.TTFFx	DC DC	Ø						
205	DC	1.SA.TTFFx	DC DC		Ø					
206	DC	1.SA.TTFFx	DC DC					Ø		
207	DC	1.1.SA.x.x	DC DC							
208	DC	1.1.SA.x.x	DC DC							
209	DC	1.1.SA.x.x	DC DC							
210	DC	1.1.SA.TFFx	DC DC							
211	DC	1.1.SA.x.x	DC DC							
212	DC	1.1.SA.x.x	DC DC							
213	Ø	DC	DC DC							
214	DC	1.SA.TTFFx	DC DC	Ø						
215	DC	1.SA.TTFFx	DC DC		Ø					
216	DC	1.SA.TTFFx	DC DC					Ø		
217	DC	1.1.SA.x.x	DC DC							
218	DC	1.1.SA.x.x	DC DC							
219	DC	1.1.SA.x.x	DC DC							
220	DC	1.1.SA.FFTx	DC DC							
221	DC	1.1.SA.x.x	DC DC							
222	DC	1.1.SA.x.x	DC DC							
223	Ø	DC	DC DC							
224	DC	1.SA.TTFFx	DC DC	Ø						

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbufo (1,2)	sts_ msgo	private_ msgo	xmit_priv ate_logical_ l_msgo	xmit_clear logical_ msgo	xmit_clear_ msgo	xmit_private_ map_msgo	xmit_priv ivate_ msgo	ftam_ msgo
225 DC	1.SA.TTFT.F1	DC DC			Ø					
226 DC	1.SA.TTFT.F1	DC DC						Ø		
227 DC	1.1.SA.1.1	DC DC								
228 DC	1.1.SA.1.1	DC DC								
229 DC	1.1.SA.1.1	DC DC								
230 DC	1.1.SA.FFT.1	DC DC								
231 DC	1.1.SA.1.1	DC DC								
232 DC	1.1.SA.1.1	DC DC								
233 Ø	DC	DC DC								
234 DC	1.SA.TTFT.F1	DC DC		Ø						
235 DC	1.SA.TTFT.F1	DC DC			Ø					
236 DC	1.SA.TTFT.F1	DC DC						Ø		
237 DC	1.1.SA.1.1	DC DC								
238 DC	1.1.SA.1.1	DC DC								
239 DC	1.1.SA.1.1	DC DC								
240 DC	1.1.SA.1.1	DC DC								
241 DC	1.1.SA.1.1	DC DC								
242 DC	1.1.SA.TFF.F1	DC DC								
243 DC	1.1.SA.1.1	DC DC								
244 DC	1.1.SA.1.1	DC DC								
245 Ø	DC	DC DC								
246 DC	1.SA.TTFT.F1	DC DC		Ø						
247 DC	1.SA.TTFT.F1	DC DC			Ø					
248 DC	1.SA.TTFT.F1	DC DC						Ø		
249 DC	1.1.SA.1.1	DC DC								
250 DC	1.1.SA.1.1	DC DC								
251 DC	1.1.SA.1.1	DC DC								
252 DC	1.1.SA.1.1	DC DC								
253 DC	1.1.SA.1.1	DC DC								
254 DC	1.1.SA.FTF.F1	DC DC								
255 DC	1.1.SA.1.1	DC DC								
256 DC	1.1.SA.1.1	DC DC								

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbufo (1,2)	sts_ msgo	private_ msgo	xmit_priv ate_logica l_msgo	xmit_clear logical_ msgo	xmit_clear_ msgo	xmit_ private_ map_msgo	xmit_priv ivate_ msgo	ftam_ msgo
257	Ø	DC DC								
258	DC	1.SA.TTFF1x	DC DC	Ø						
259	DC	1.SA.TTFF1x	DC DC		Ø					
260	DC	1.SA.TTFF1x	DC DC					Ø		
261	DC	1.1.SA.1x	DC DC							
262	DC	1.1.SA.1x	DC DC							
263	DC	1.1.SA.1x	DC DC							
264	DC	1.1.SA.1x	DC DC							
265	DC	1.1.SA.1x	DC DC							
266	DC	1.1.SA.FF1x	DC DC							
267	DC	1.1.SA.1x	DC DC							
268	DC	1.1.SA.1x	DC DC							
269	Ø	DC	DC DC							
270	DC	1.SA.TTFF1x	DC DC	Ø						
271	DC	1.SA.TTFF1x	DC DC						Ø	Ø
272	DC	1.1.SA.1x	DC DC							
273	DC	1.1.SA.1x	DC DC							
274	DC	1.1.SA.1x	DC DC							
275	DC	1.1.SA.TTFF1x	DC DC							
276	DC	1.1.SA.1x	DC DC							
277	DC	1.1.SA.1x	DC DC							
278	Ø	DC	DC DC							
279	DC	1.SA.TTFF1x	DC DC	Ø						
280	DC	1.SA.TTFF1x	DC DC						Ø	
281	DC	1.1.SA.1x	DC DC							Ø
282	DC	1.1.SA.1x	DC DC							
283	DC	1.1.SA.1x	DC DC							
284	DC	1.1.SA.TTFF1x	DC DC							
285	DC	1.1.SA.1x	DC DC							
286	DC	1.1.SA.1x	DC DC							
287	Ø	DC	DC DC							
288	Ø	1.SA.TTFF1x	DC DC	Ø						

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfero	outbufo (1,2)	sts_ msgo	private_ msgo	xmit_priv ate_logica l_msgo	xmit_clear logical_ msgo	xmit_clear_ msgo	xmit_ private_ map_msgo	xmit_priv ivate_ msgo	ftam_ msgo
289 DC	1.SA.TEFF ₁	DC DC							Ø	Ø
290 DC	1.1.SA.1.1	DC DC								
291 DC	1.1.SA.1.1	DC DC								
292 DC	1.1.SA.1.1	DC DC								
293 DC	1.1.SA.FFT ₁	DC DC								
294 DC	1.1.SA.1.1	DC DC								
295 DC	1.1.SA.1.1	DC DC								
296 Ø	DC	DC DC								
297 DC	1.SA.TEFF ₁	DC DC		Ø						
298 DC	1.SA.TEFF ₁	DC DC							Ø	
299 DC	1.1.SA.1.1	DC DC								
300 DC	1.1.SA.1.1	DC DC								
301 DC	1.1.SA.1.1	DC DC								
302 DC	1.1.SA.TEFF ₁	DC DC								
303 DC	1.1.SA.1.1	DC DC								
304 DC	1.1.SA.1.1	DC DC								
305 Ø	DC	DC DC								
306 DC	1.SA.TEFF ₁	DC DC		Ø						
307 DC	1.SA.TEFF ₁	DC DC							Ø	
308 DC	1.1.SA.1.1	DC DC								
309 DC	1.1.SA.1.1	DC DC								
310 DC	1.1.SA.1.1	DC DC								
311 DC	1.1.SA.FT ₁	DC DC								
312 DC	1.1.SA.1.1	DC DC								
313 DC	1.1.SA.1.1	DC DC								
314 Ø	DC	DC DC								
315 DC	1.SA.TEFF ₁	DC DC		Ø						
316 DC	1.SA.TEFF ₁	DC DC							Ø	
317 DC	1.1.SA.1.1	DC DC								
318 DC	1.1.SA.1.1	DC DC								
319 DC	1.1.SA.1.1	DC DC								
320 DC	1.1.SA.FFT ₁	DC DC								

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbu ₀ (1,2)	sts_ msg ₀	private_ msg ₀	xmit_priv ate logical l_msg ₀	xmit_clear logical_ msg ₀	xmit_clear_ map_ msg ₀	xmit_ private_ map_ msg ₀	xmit_priv ivate_ msg ₀	stam_ msg ₀
321	DC	1,1,SA,1,1	DC DC							
322	DC	1,1,SA,1,1	DC DC							
323	Ø	DC	DC DC							
324	DC	1,SA,TFFFT,1	DC DC	Ø					Ø	
325	DC	1,SA,TFFFT,1	DC DC							
326	DC	1,1,SA,1,1	DC DC							
327	DC	1,1,SA,1,1	DC DC							
328	DC	1,1,SA,1,1	DC DC							
329	DC	1,1,SA,1,1	DC DC							
330	DC	1,1,SA,1,1	DC DC							
331	DC	1,1,SA,TFF,1	DC DC							
332	DC	1,1,SA,1,1	DC DC							
333	DC	1,1,SA,1,1	DC DC							
334	Ø	DC	DC DC							
335	DC	1,SA,TFFFT,1	DC DC	Ø					Ø	
336	DC	1,SA,TFFFT,1	DC DC							
337	DC	1,1,SA,1,1	DC DC							
338	DC	1,1,SA,1,1	DC DC							
339	DC	1,1,SA,1,1	DC DC							
340	DC	1,1,SA,1,1	DC DC							
341	DC	1,1,SA,1,1	DC DC							
342	DC	1,1,SA,TFF,1	DC DC							
343	DC	1,1,SA,1,1	DC DC							
344	DC	1,1,SA,1,1	DC DC							
345	Ø	DC	DC DC							
346	DC	1,SA,TFFFT,1	DC DC	Ø					Ø	
347	DC	1,SA,TFFFT,1	DC DC							
348	DC	1,1,SA,1,1	DC DC							
349	DC	1,1,SA,1,1	DC DC							
350	DC	1,1,SA,1,1	DC DC							
351	DC	1,1,SA,1,1	DC DC							
352	DC	1,1,SA,1,1	DC DC							

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

MEDIUM ₀	Transfer ₀	outbufo (1,2)	sts_ msgo	private_ msgo	xmit_priv ate logical l_msgo	xmit_clear logical_ msgo	xmit_clear_ msgo	xmit_private map msgo	xmit_private_ msgo	stam_ msgo
353	DC	1.1.SAFFT.1	DC	DC						
354	DC	1.1.SA.1.1	DC	DC						
355	DC	1.1.SA.1.1	DC	DC						
356	Ø	DC	DC	DC						
357										
358										
359										
360										
361										
362										
363										
364										
365										
366										
367										
368										
369										
370										
371										
372										
373										
374										
375										
376										
377										
378										
379										
380										
381										
382										
383										
384										

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase_ msgO	cmise_ msgO	rose_ msgO	presen tation_ msgO	session_ msgO	co trans_ msgO	cl_trans_ msgO	es/is_ msgO	is/is_ msgO	clnp_ msgO	llc_ msgO	inbuf	SE
1												1,1,SA,1,1	1
2													0
3													2
4													3
5													6
6							Ø						17
7										Ø	Ø		18
8													0
9													9
10													14
11				Ø									15
12					Ø								16
13						Ø							4
14								Ø					17
15										Ø	Ø		18
16													0
17													9
18													14
19			Ø										15
20					Ø								16
21						Ø							5
22									Ø				17
23											Ø		18
24													0
25													9
26													14
27			Ø										15
28					Ø								16
29						Ø							6
30										Ø			17
31											Ø		18
32													0

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase msgO	cmise msgO	rose msgO	presen- tation- msgO	session- msgO	co trans- msgO	cl_trans- msgO	es/is- msgO	is/is- msgO	clnp- msgO	llc- msgO	inbuf	Se
33													10
34	Ø												14
35				Ø									15
36					Ø								16
37						Ø							4
38								Ø					17
39											Ø		18
40													0
41													10
42	Ø												14
43				Ø									15
44					Ø								16
45						Ø							5
46									Ø				17
47											Ø		18
48													0
49													10
50	Ø												14
51				Ø									15
52					Ø								16
53						Ø							6
54										Ø			17
55											Ø		18
56													0
57													11
58	Ø												12
59		Ø											13
60			Ø										14
61				Ø									15
62					Ø								16
63						Ø							4
64								Ø					17

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msg()	smase_ msg()	cmise_ msg()	rose_ msg()	presen tation_ msg()	session_ msg()	co trans_ msg()	cl trans_ msg()	es/is_ msg()	is/is_ msg()	clnp_ msg()	llc_ msg()	inbuf	SE
65											Ø		18
66													0
67													11
68	Ø												12
69		Ø											13
70			Ø										14
71				Ø									15
72					Ø								16
73						Ø							5
74									Ø				17
75											Ø		18
76													0
77													11
78	Ø												12
79		Ø											13
80			Ø										14
81				Ø									15
82					Ø								16
83						Ø							6
84										Ø			17
85											Ø		18
86													0
87													8
88													9
89													14
90				Ø									15
91					Ø								16
92						Ø							4
93								Ø					17
94											Ø		18
95													0
96													8

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase msgO	cmise msgO	rose msgO	presen tation msgO	session msgO	co trans msgO	cl trans msgO	es/is msgO	is/is msgO	clnp msgO	llc msgO	inbuf	SE
97													9
98													14
99				Ø									15
100					Ø								16
101						Ø							5
102									Ø				17
103											Ø		18
104													0
105													8
106													9
107													14
108				Ø									15
109					Ø								16
110						Ø							6
111										Ø			17
112											Ø		18
113													0
114													8
115													10
116 Ø													14
117				Ø									15
118					Ø								16
119						Ø							4
120								Ø					17
121											Ø		18
122													0
123													8
124													10
125 Ø													14
126				Ø									15
127					Ø								16
128						Ø							5

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase- msgO	cmise- msgO	rose- msgO	presen- tation- msgO	session- msgO	co trans- msgO	cl_trans- msgO	es/is- msgO	is/is- msgO	clnp- msgO	llc- msgO	inbuf	Se
129									Ø				17
130											Ø		18
131													0
132													8
133													10
134	Ø												14
135				Ø									15
136					Ø								16
137													6
138										Ø			17
139											Ø		18
140													0
141													8
142													11
143	Ø												12
144		Ø											13
145			Ø										14
146				Ø									15
147					Ø								16
148						Ø							4
149								Ø					17
150											Ø		18
151													0
152													8
153													11
154	Ø												12
155		Ø											13
156			Ø										14
157				Ø	Ø								15
158													16
159						Ø							5
160									Ø				17

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase msgO	cmise msgO	rose msgO	presen tation msgO	session msgO	co trans msgO	cl trans msgO	es/is msgO	is/is msgO	clnp_ msgO	llc_ msgO	inbuf	SE
161											Ø		18
162													0
163													8
164													11
165	Ø												12
166		Ø											13
167			Ø										14
168				Ø									15
169					Ø								16
170						Ø							6
171										Ø			17
172											Ø		18
173													0
174													7
175													8
176													9
177													14
178				Ø									15
179					Ø								16
180						Ø							4
181								Ø					17
182											Ø		18
183													0
184													7
185													8
186													9
187													14
188				Ø									15
189					Ø								16
190						Ø							5
191									Ø				17
192											Ø		18

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase msgO	cmise msgO	rose msgO	presen tation msgO	session msgO	co trans msgO	cl_trans msgO	es/is msgO	is/is msgO	clnp_ msgO	llc_ msgO	inbuf	Se
193													0
194													7
195													8
196													9
197													14
198				Ø									15
199					Ø								16
200						Ø							6
201										Ø			17
202											Ø		18
203													0
204													7
205													8
206													10
207	Ø												14
208				Ø									15
209					Ø								16
210						Ø							4
211								Ø					17
212											Ø		18
213													0
214													7
215													8
216													10
217	Ø												14
218				Ø									15
219					Ø								16
220						Ø							5
221									Ø				17
222											Ø		18
223													0
224													7

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase msgO	cmise msgO	prese- ntation msgO	session msgO	co trans- msgO	cl trans- msgO	es/is- msgO	is/is- msgO	chnp- msgO	llc- msgO	inbuf	SE
225												8
226												10
227	Ø											14
228			Ø									15
229				Ø								16
230					Ø							6
231									Ø			17
232										Ø		18
233												0
234												7
235												8
236												11
237	Ø											12
238		Ø										13
239			Ø									14
240				Ø								15
241					Ø							16
242						Ø						4
243							Ø					17
244										Ø		18
245												0
246												7
247												8
248												11
249	Ø											12
250		Ø										13
251			Ø									14
252				Ø								15
253					Ø							16
254						Ø						5
255								Ø				17
256										Ø		18

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase_ msgO	cmise_ msgO	rose_ msgO	presen tation_ msgO	session_ msgO	co trans_ msgO	cl_trans_ msgO	es/is_ msgO	is/is_ msgO	clnp_ msgO	llc_ msgO	inbuf	Se
257													0
258													7
259													8
260													11
261	Ø												12
262		Ø											13
263			Ø										14
264				Ø									15
265					Ø								16
266						Ø							6
267										Ø			17
268											Ø		18
269													0
270													7
271													9
272													14
273				Ø									15
274					Ø								16
275						Ø							4
276								Ø					17
277											Ø		18
278													0
279													7
280													9
281													14
282				Ø									15
283					Ø								16
284						Ø							5
285									Ø				17
286											Ø		18
287													0
288													7

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase msgO	cmise msgO	rose msgO	presen tation msgO	session msgO	co_trans msgO	cl_trans msgO	es/is msgO	is/is msgO	clnp_ msgO	llc_ msgO	inbuf	SE
289													9
290													14
291				Ø									15
292					Ø								16
293						Ø							6
294										Ø			17
295											Ø		18
296												0	7
297													10
298													14
299	Ø												15
300				Ø									16
301					Ø								4
302						Ø							17
303								Ø					18
304											Ø		0
305													7
306													10
307													14
308	Ø												15
309				Ø									16
310					Ø								5
311						Ø							17
312									Ø				18
313											Ø		0
314													7
315													10
316													14
317	Ø												15
318				Ø									16
319					Ø								6
320						Ø							

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase msgO	cmise msgO	rose msgO	presen tation msgO	session msgO	co trans msgO	cl_trans_ msgO	es/is_ msgO	is/is_ msgO	clap_ msgO	llc_ msgO	inbuf	Se
321										Ø			17
322											Ø		18
323													0
324													7
325													11
326	Ø												12
327		Ø											13
328			Ø										14
329				Ø									15
330					Ø								16
331						Ø							4
332								Ø					17
333											Ø		18
334													0
335													7
336													11
337	Ø												12
338		Ø											13
339			Ø										14
340				Ø									15
341					Ø								16
342						Ø							5
343									Ø				17
344											Ø		18
345													0
346													7
347													11
348	Ø												12
349		Ø											13
350			Ø										14
351				Ø									15
352					Ø								16

Table 2: TEST SEQUENCE FOR THE OSI PROFILE (CONTINUED)

acse msgO	smase msgO	cmise msgO	prese- ntation msgO	session msgO	co trans msgO	cl trans msgO	es/is msgO	is/is msgO	clnp - msgO	llc - msgO	inbuf	Se
353					Ø							6
354									Ø			17
355										Ø		18
356												0
357												
358												
359												
360												
361												
362												
363												
364												
365												
366												
367												
368												
369												
370												
371												
372												
373												
374												
375												
376												
377												
378												
379												
380												
381												
382												
383												
384												

LIST OF REFERENCES

- [ANNA88] Annamalai, K., "FDDI Physical Layer Implementation Considerations," Proceedings of SPIE Fiber Optic Datacom and Computer Networks, vol. 991, International Society for Optical Engineering, Bellingham, WA., 1988.
- [FINL84] Finley, M.R., "Optical Fibers in Local Area Communications," IEEE Communication Magazine, Volume 22, No. 8, Aug. 1984.
- [GREE89] Green, D.T. and D.T. Marlow, "SAFENET - A LAN for Navy Mission Critical Systems," Proc. 14th Conf. Local Computer Networks, Minneapolis, MN, pp. 340-346, Oct. 1989.
- [HDBK92] Military Handbook, "Survivable Adaptable Fiber Optic Embedded Network (SAFENET)," Sep. 1992, Naval Ocean Systems Center, MIL-HDBK-2204.
- [IFOC83] International Fiber Optics and Communications 1983-84 Handbook and Buyers Guide, Volume V, published by Information Gatekeepers, Inc.
- [ISO8073] ISO 8073, "Information Processing Systems - Open Systems Interconnection-Connection Oriented Transport Protocol Specification," 1986.
- [JOHN87] Johnson, R.A. and R.C. Stewart, "Implementation of Fiber Optics in U.S. Naval Combatants," Proc. SPIE, vol 840, pp. 80-93, Aug. 20-21 1987.
- [KIMB89] Kimball, Robert M., "Optical Performance models for FDDI Links," Proceedings of SPIE Fiber Networking and Telecommunications, vol. 1179, International Society for Optical Engineering, Bellingham, WA., 1989.
- [KOCH91] Kochanski, R.J. and J.L. Paige, "SAFENET: The Standard and Its Application," IEEE LCS, Feb. 1991, Vol. 2, No. 1, pp. 46-51.
- [LI1983] Li, T., "Advances in optical fiber communications: an historical perspective," IEEE J. Select. Areas commun., SAC - 1, no. 3, pp. 356-372, Apr. 1983.
- [LUND91] Lundy, G.M., "Improving Throughput in the FDDI Token Ring Network," Proc. of the Second International Workshop on Protocols for High-Speed Networks, IFIP, North Holland, 1991.
- [MILL90] Miller, R.E. and G.M. Lundy, "Testing Protocol Implementations Based on a Formal Specification," Proc. of the Third International Workshop on Protocol Test Systems, IFIP, North Holland, Oct. 1990.

- [PAIG90] Paige, J.L. and E.A. Howard "SAFENET II - The Navy's - FDDI based computer network standard," Proceedings of SPIE Campus-Wide, and Metropolitan Area Networks, vol 1364, International Society for Optical Engineering, Bellingham, WA., 1990.
- [ROSS89] Ross, F.E., "An Overview of FDDI: The Fiber Distributed Data Interface," IEEE Journal on Selected Areas in Communications, Sep. 1989.
- [STAL88] Stallings, William, Data and Computer Communications, Macmillan Publishing Company, New York, NY, 1988.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
Dudley Knox Library Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
Chairman, Code CS Computer Science Department Naval Postgraduate School Monterey, CA 93943	2
Dr. G. M. Lundy Computer Science Department, Code CSLn Naval Postgraduate School Monterey, CA 93943	3
Prof. L. Stevens Computer Science Department, Code CSSt Naval Postgraduate School Monterey, CA 93943	1
LT. Wayne High Rte. 1 Box 99A Eutawville, SC 29048	2
CDR. Greg Sawyer SPAWAR SYSCOM, Code 2312 5 Crystal Park Washington, DC 20363	1